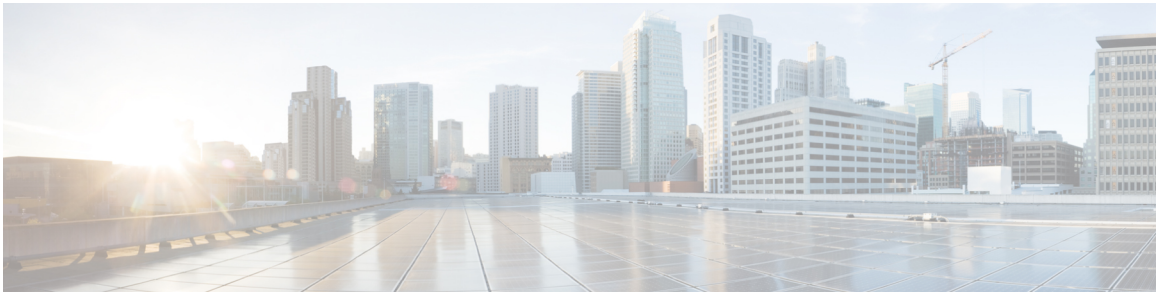
Security and VPN Configuration Guide, Cisco IOS XE 17.x

First Published: 2021-01-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

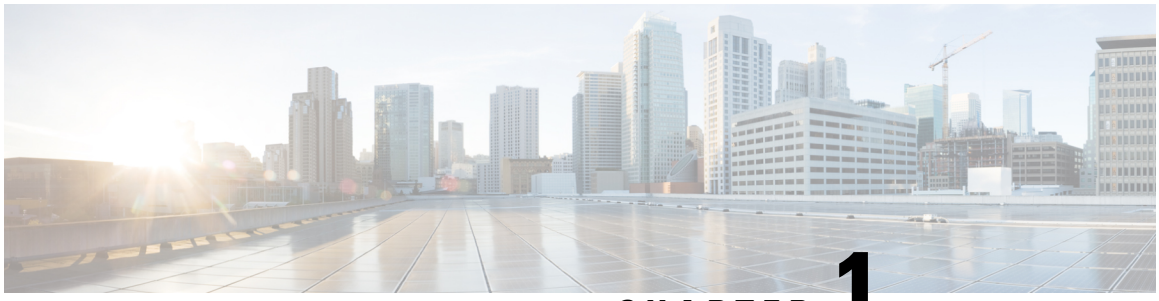
Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

Authentication Authorization and Accounting



CHAPTER 1

Configuring Authentication

Authentication provides a method to identify users, which includes the login and password dialog, challenge and response, messaging support, and encryption, depending on the selected security protocol. Authentication is the way a user is identified prior to being allowed access to the network and network services.

Prerequisites for Configuring Authentication

The Cisco IOS XE implementation of authentication is divided into AAA Authentication and non-authentication methods. Cisco recommends that, whenever possible, AAA security services be used to implement authentication.

Restrictions for Configuring Authentication

- The number of AAA method lists that can be configured is 250.
- Web authentication is not supported on Cisco IOS XE software.

Information About Configuring Authentication

The following sections describe how AAA authentication is configured by defining a named list of authentication methods and then applying that list to various interfaces. This section also describes how AAA authentication is handled by using RADIUS Change in Authorization (CoA):

Named Method Lists for Authentication

To configure AAA authentication, you must first define a named list of authentication methods, and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS XE software uses the first listed method to authenticate users. If that method fails to respond, the Cisco IOS XE software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or all methods defined in the method list are exhausted.

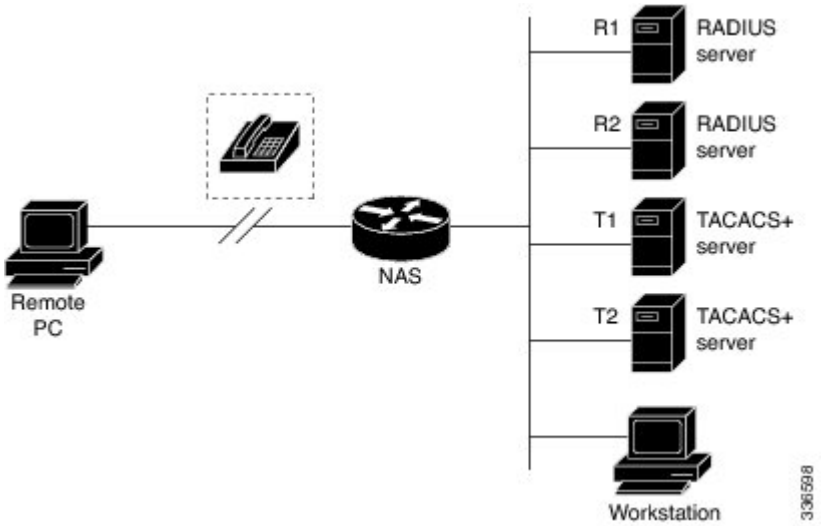
It is important to note that the Cisco IOS XE software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle--meaning that the security server or local username database responds by denying the user access--the authentication process stops and no other authentication methods are attempted.

Note The number of AAA method lists that can be configured is 250.

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 1: Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as a server group, and define T1 and T2 as a separate server group. For example, you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, authentication--the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, refer to the “Configuring RADIUS” or “Configuring TACACS+” chapter.

Method List Examples

Suppose the system administrator has decided on a security solution where all interfaces will use the same authentication methods to authenticate PPP connections. In the RADIUS group, R1 is contacted first for authentication information, then if there is no response, R2 is contacted. If R2 does not respond, T1 in the TACACS+ group is contacted; if T1 does not respond, T2 is contacted. If all designated servers fail to respond, authentication falls to the local username database on the access server itself. To implement this solution, the system administrator would create a default method list by entering the following command:

```
aaa authentication ppp default group radius group tacacs+ local
```

In this example, “default” is the name of the method list. The protocols included in this method list are listed after the name, in the order they are to be queried. The default list is automatically applied to all interfaces.

When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated.

It is important to remember that a FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Suppose the system administrator wants to apply a method list only to a particular interface or set of interfaces. In this case, the system administrator creates a named method list and then applies this named list to the applicable interfaces. The following example shows how the system administrator can implement an authentication method that will be applied only to interface 3:

```
aaa authentication ppp default group radius group tacacs+ local
aaa authentication ppp apple group radius group tacacs+ local none
 interface async 3
 ppp authentication chap apple
```

In this example, “apple” is the name of the method list, and the protocols included in this method list are listed after the name in the order in which they are to be performed. After the method list has been created, it is applied to the appropriate interface. Note that the method list name (apple) in both the AAAand PPP authentication commands must match.

In the following example, the system administrator uses server groups to specify that only R2 and T2 are valid servers for PPP authentication. To do this, the administrator must define specific server groups whose members are R2 (172.16.2.7) and T2 (172.16.2.77), respectively. In this example, the RADIUS server group “rad2only” is defined as follows using the **aaa group server** command:

```
aaa group server radius rad2only
 server 172.16.2.7
```

The TACACS+ server group “tac2only” is defined as follows using the **aaa group server** command:

```
aaa group server tacacs+ tac2only
 server 172.16.2.77
```

The administrator then applies PPP authentication using the server groups. In this example, the default methods list for PPP authentication follows this order: **group rad2only**, **group tac2only**, and **local**:

```
aaa authentication ppp default group rad2only group tac2only local
```

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. The Cisco software supports the RADIUS CoA request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Use the following per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce
- Security and Password
- Accounting

CoA Requests

CoA requests, as described in RFC 5176, are used in a pushed model to allow for session identification, host reauthentication, and session termination. The model comprises one request (CoA-Request) and two possible response codes:

- CoA acknowledgment (ACK) [CoA-ACK]
- CoA non-acknowledgment (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the device that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the device for a session termination.

The following table shows the IETF attributes that are supported for the RADIUS Change of Authorization (CoA) feature.

Table 1: Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

The following table shows the possible values for the Error-Cause attribute.

Table 2: Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable

Value	Explanation
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request Response code can be used to issue a command to the device. The supported commands are listed in the "CoA Request Commands" section.

The packet format for a CoA Request Response code as defined in RFC 5176 consists of the following fields: Code, Identifier, Length, Authenticator, and Attributes in the Type:Length:Value (TLV) format.

The Attributes field is used to carry Cisco VSAs.

Session Identification

For disconnect and CoA requests targeted at a particular session, the device locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)
- Audit-Session-Id (Cisco vendor-specific attribute (VSA))
- Calling-Station-Id (IETF attribute #31, which contains the host MAC address)

Unless all session identification attributes included in the CoA message match the session, the device returns a Disconnect-NAK or CoA-NAK with the "Invalid Attribute Value" error-code attribute.

Note A CoA NAK message is not sent for all CoA requests with a key mismatch. The message is sent only for the first three requests for a client. After that, all the packets from that client are dropped. When there is a key mismatch, the response authenticator sent with the CoA NAK message is calculated from a dummy key value.

CoA ACK Response Code

If an authorization state is changed successfully, a positive acknowledgment (ACK) is sent. The attributes returned within a CoA ACK can vary based on the CoA Request.

CoA NAK Response Code

A negative acknowledgment (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure.

CoA Request Commands

The commands supported on the device are shown in the table below. All CoA commands must include the session identifier between the device and the CoA client.

Table 3: CoA Request Commands Supported on the Device

Command	Cisco VSA
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the device's response to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1*x*, the device responds by sending an Extensible Authentication Protocol over LAN (EAPoL)-RequestId message to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network. If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism.

CoA Request Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism. This command is carried in a standard CoA-Request message that has the following VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section. If the device cannot locate the session, it returns

a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates the session, it disables the hosting port and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

To ignore the RADIUS server CoA disable port command, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

CoA Request Bounce Port

A RADIUS server CoA bounce port sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer) that does not have a mechanism to detect a change on this authentication port. The CoA bounce port is carried in a standard CoA-Request message that contains the following VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the Session Identification. If the session cannot be located, the device returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device disables the hosting port for a period of 10 seconds, reenables it (port-bounce), and returns a CoA-ACK.

To ignore the RADIUS server CoA bounce port, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

Domain Stripping

You can remove the domain name from the username received at the global level by using the **radius-server domain-stripping** command. When the **radius-server domain-stripping** command is configured, all the AAA requests with "user@example.com" go to the remote RADIUS server with the reformatted username "user." The domain name is removed from the request.

Note Domain stripping will not be done in a TACACS configuration.

The AAA Broadcast Accounting feature allows accounting information to be sent to multiple AAA servers at the same time, that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows you to send accounting information to private and public AAA servers. It also provides redundant billing information for voice applications.

The Domain Stripping feature allows domain stripping to be configured at the server group level.

Per-server group configuration overrides the global configuration. If domain stripping is not enabled globally, but it is enabled in a server group, then it is enabled only for that server group. Also, if virtual routing and forwarding (VRF)-specific domain stripping is configured globally and in a server group for a different VRF, domain stripping is enabled in both the VRFs. VRF configurations are taken from server-group configuration mode. If server-group configurations are disabled in global configuration mode but are available in server-group configuration mode, all configurations in server-group configuration mode are applicable.

After the domain stripping and broadcast accounting are configured, you can create separate accounting records as per the configurations.

How to Configure AAA Authentication Methods

Note AAA features are not available until you enable AAA globally using the **aaa new-model** command.

For authentication configuration examples using the commands in this chapter, refer to the Authentication Examples.

Configuring Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login**command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login**command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication login**{**default** | *list-name*} *method1*[*method2...*]
3. Router(config)# **line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**]
4. Router(config-line)# **login authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication login**{**default** | *list-name*} *method1*[*method2...*]	Creates a local authentication list.
Step 3	Router(config)# **line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**]	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 4	Router(config-line)# **login authentication** **Example:** {**default** | *list-name*}	Applies the authentication list to a line or set of lines.

What to do next

The *list-name* is a character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if

the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication login default group tacacs+ none
```

Note Because the **none** keyword enables *any* user logging in to successfully authenticate, it should be used only as a backup method of authentication.

To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa authentication login default group radius
```

The table below lists the supported login authentication methods.

Table 4: AAA Authentication Login Methods

Keyword	Description
enable	Uses the enable password for authentication.
krb5	Uses Kerberos 5 for authentication.
krb5-telnet	Uses Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. If selected, this keyword must be listed as the first method in the method list.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Note The **login** command only changes username and privilege level but does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

Login Authentication Using Enable Password

Use the **aaa authentication login**command with the **enable** *method* keyword to specify the enable password as the login authentication method. For example, to specify the enable password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

Login Authentication Using Kerberos

Authentication via Kerberos is different from most other authentication methods: the user's password is never sent to the remote access server. Remote users logging in to the network are prompted for a username. If the key distribution center (KDC) has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user's credential cache on the router.

While krb5 does use the KINIT program, a user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS XE implementation of Kerberos.

Use the **aaa authentication login**command with the **krb5***method* keyword to specify Kerberos as the login authentication method. For example, to specify Kerberos as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default krb5
```

Before you can use Kerberos as the login authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter "Configuring Kerberos."

Login Authentication Using Line Password

Use the **aaa authentication login**command with the **line** *method* keyword to specify the line password as the login authentication method. For example, to specify the line password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password. For more information about defining line passwords, refer to the Configuring Line Password Protection.

Login Authentication Using Local Password

Use the **aaa authentication login**command with the **local** *method* keyword to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default local
```

For information about adding users into the local username database, refer to the Establishing Username Authentication.

Login Authentication Using Group RADIUS

Use the **aaa authentication login**command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group radius
```

Before you can use RADIUS as the login authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter “Configuring RADIUS.”

Configuring RADIUS Attribute 8 in Access Requests

After you have used the **aaa authentication login** command to specify RADIUS and your login host has been configured to request its IP address from the NAS, you can send attribute 8 (Framed-IP-Address) in access-request packets by using the **radius-server attribute 8 include-in-access-req** command in global configuration mode. This command makes it possible for NAS to provide the RADIUS server a hint of the user IP address in advance for user authentication. For more information about attribute 8, refer to the appendix “RADIUS Attributes” at the end of the book.

Login Authentication Using Group TACACS

Use the **aaa authentication login**command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group tacacs+
```

Before you can use TACACS+ as the login authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter “Configuring TACACS+.”

Login Authentication Using group group-name

Use the **aaa authentication login**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
```

```
server 172.16.2.3
server 172.16.2 17
server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group loginrad
```

Before you can use a group name as the login authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring PPP Authentication Using AAA

Many users access network access servers through dialup via async or ISDN. Dialup via async or ISDN bypasses the CLI completely; instead, a network protocol (such as PPP or ARA) starts as soon as the connection is established.

The AAA security services facilitate a variety of authentication methods for use on serial interfaces running PPP. Use the **aaa authentication ppp**command to enable AAA authentication no matter which of the supported PPP authentication methods you decide to use.

To configure AAA authentication methods for serial lines using PPP, use the following commands in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa new-model**
2. Router(config)# **aaa authentication ppp**{**default** | *list-name*} *method1*[*method2...*]
3. Router(config)# **interface** *interface-type interface-number*
4. Router(config-if)# **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**][**optional**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa new-model**	Enables AAA globally.
Step 2	Router(config)# **aaa authentication ppp**{**default** | *list-name*} *method1*[*method2...*]	Creates a local authentication list.
Step 3	Router(config)# **interface** *interface-type interface-number*	Enters interface configuration mode for the interface to which you want to apply the authentication list.
Step 4	Router(config-if)# **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**][**optional**]	Applies the authentication list to a line or set of lines. In this command, *protocol1* and *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by *protocol1*. If *protocol1* is unable to

	Command or Action	Purpose
		establish authentication, the next configured protocol is used to negotiate authentication.

What to do next

With the **aaa authentication ppp**command, you create one or more lists of authentication methods that are tried when a user tries to authenticate via PPP. These lists are applied using the **ppp authentication** line configuration command.

To create a default list that is used when a named list is *not* specified in the **ppp authentication** command, use the **default** keyword followed by the methods you want used in default situations.

For example, to specify the local username database as the default method for user authentication, enter the following command:

```
aaa authentication ppp default local
```

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication ppp default group tacacs+ none
```

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

The table below lists the supported login authentication methods.

Table 5: AAA Authentication PPP Methods

Keyword	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
krb5	Uses Kerberos 5 for authentication (can only be used for PAP authentication).
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.

Keyword	Description
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

PPP Authentication Using Kerberos

Use the **aaa authentication ppp**command with the **krb5***method* keyword to specify Kerberos as the authentication method for use on interfaces running PPP. For example, to specify Kerberos as the method of user authentication when no other method list has been defined, enter the following command:

```
aaa authentication ppp default krb5
```

Before you can use Kerberos as the PPP authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter “Configuring Kerberos”.

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

Use the **aaa authentication ppp**command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of authentication for use on lines running PPP when no other method list has been defined, enter the following command:

```
aaa authentication ppp default local
```

For information about adding users into the local username database, refer to the Establishing Username Authentication.

PPP Authentication Using Group RADIUS

Use the **aaa authentication ppp**command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group radius
```

Before you can use RADIUS as the PPP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter “Configuring RADIUS.”

Configuring RADIUS Attribute 44 in Access Requests

After you have used the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method, you can configure your device to send attribute 44 (Acct-Session-ID) in access-request packets by using the **radius-server attribute 44 include-in-access-req** command in global configuration mode. This command allows the RADIUS daemon to track a call from the beginning to the end.

PPP Authentication Using Group TACACS

Use the **aaa authentication ppp**command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group tacacs+
```

Before you can use TACACS+ as the PPP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

PPP Authentication Using group group-name

Use the **aaa authentication ppp**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group ppprad**:

```
aaa group server radius ppprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *ppprad*.

To specify **group ppprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group ppprad
```

Before you can use a group name as the PPP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

To allocate a specific number of background processes to handle AAA requests for PPP, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa processes number`	Allocates a specific number of background processes to handle AAA authentication and authorization requests for PPP.

The argument *number* defines the number of background processes earmarked to process AAA authentication and authorization requests for PPP and can be configured for any value from 1 to 2147483647. Because of

the way the PPP manager handles requests for PPP, this argument also defines the number of new users that can be simultaneously authenticated. This argument can be increased or decreased at any time.

Note Allocating additional background processes can be expensive. You should configure the minimum number of background processes capable of handling the AAA requests for PPP.

Configuring ARAP Authentication Using AAA

Using the **aaa authentication arap** command, you can create one or more lists of authentication methods that are tried when AppleTalk Remote Access Protocol (ARAP) users attempt to log in to the device. These lists are used with the **arap authentication** line configuration command.

Use the following commands starting in global configuration mode:

SUMMARY STEPS

1. Device(config)# **aaa new-model**
2. Device(config)# **aaa authentication arap**
3. Device(config)# **line** *number*
4. Device(config-line)# **autoselect arap**
5. Device(config-line)# **autoselect during-login**
6. Device(config-line)# **arap authentication** *list-name*
7. Device(config-line)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **aaa new-model**	Enables AAA globally.
Step 2	Device(config)# **aaa authentication arap** **Example:** Enables authentication for ARAP users.	
Step 3	Device(config)# **line** *number*	(Optional) Changes to line configuration mode.
Step 4	Device(config-line)# **autoselect arap**	(Optional) Enables autoselection of ARAP.
Step 5	Device(config-line)# **autoselect during-login**	(Optional) Starts the ARAP session automatically at user login.
Step 6	Device(config-line)# **arap authentication** *list-name*	(Optional—not needed if **default** is used in the **aaa authentication arap** command) Enables TACACS+ authentication for ARAP on a line.
Step 7	Device(config-line)# **end**	Returns to the privileged EXEC mode.

What to do next

The *list-name* is any character string used to name the list you are creating. The *method* argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **arap authentication** command, use the **default** keyword followed by the methods you want to use in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to be authenticated, it should be used as a backup method of authentication.

The following table lists the supported login authentication methods.

Table 6: AAA Authentication ARAP Methods

Keyword	Description
auth-guest	Allows guest logins only if the user has already logged in to EXEC mode.
guest	Allows guest logins.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

For example, to create a default AAA authentication method list used with ARAP, use the following command:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for ARAP and name the list *MIS-access*, use the following command:

```
aaa authentication arap MIS-access if-needed none
```

This section includes the following sections:

ARAP Authentication Allowing Authorized Guest Logins

Use the **aaa authentication arap**command with the **auth-guest** keyword to allow guest logins only if the user has already successfully logged in to the EXEC. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to

allow all authorized guest logins--meaning logins by users who have already successfully logged in to the EXEC--as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default auth-guest group radius
```

Note By default, guest logins through ARAP are disabled when you initialize AAA. To allow guest logins, you must use the **aaa authentication arap** command with either the **guest** or the **auth-guest** keyword.

ARAP Authentication Allowing Guest Logins

Use the **aaa authentication arap**command with the **guest** keyword to allow guest logins. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all guest logins as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default guest group radius
```

ARAP Authentication Using Line Password

Use the **aaa authentication arap**command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default line
```

Before you can use a line password as the ARAP authentication method, you need to define a line password. For more information about defining line passwords, refer to the section Configuring Line Password Protection in this chapter.

ARAP Authentication Using Local Password

Use the **aaa authentication arap**command with the *method*keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default local
```

For information about adding users to the local username database, refer to the Establishing Username Authentication.

ARAP Authentication Using Group RADIUS

Use the **aaa authentication arap**command with the **group radius** *method* to specify RADIUS as the ARAP authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group radius
```

Before you can use RADIUS as the ARAP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

ARAP Authentication Using Group TACACS

Use the **aaa authentication arap**command with the **group tacacs+** *method* to specify TACACS+ as the ARAP authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group tacacs+
```

Before you can use TACACS+ as the ARAP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

ARAP Authentication Using Group group-name

Use the **aaa authentication arap**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the ARAP authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server**command to first define the members of **group araprad**:

```
aaa group server radius araprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *araprad*.

To specify **group araprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group araprad
```

Before you can use a group name as the ARAP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring NASI Authentication Using AAA

Using the **aaa authentication nasi** command, you can create one or more lists of authentication methods that are tried when NetWare Asynchronous Services Interface (NASI) users attempt to log in to the device. These lists are used with the **nasi authentication line** configuration command.

To configure NASI authentication using AAA, use the following commands starting in global configuration mode:

SUMMARY STEPS

1. Device(config)# **aaa new-model**
2. Device(config)# **aaa authentication nasi**

3. Device(config)# **line** *number*
4. Device(config-line)# **nasi authentication** *list-name*
5. Device(config-line)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **aaa new-model**	Enables AAA globally.
Step 2	Device(config)# **aaa authentication nasi** **Example:**	Enables authentication for NASI users.
Step 3	Device(config)# **line** *number*	(Optional--not needed if **default** is used in the **aaa authentication nasi** command) Enters line configuration mode.
Step 4	Device(config-line)# **nasi authentication** *list-name*	(Optional--not needed if **default** is used in the **aaa authentication nasi** command) Enables authentication for NASI on a line.
Step 5	Device(config-line)# **end**	Returns to the privileged EXEC mode.

What to do next

The *list-name* is any character string used to name the list you are creating. The *method* argument refers to the actual list of methods that the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **aaa authentication nasi** command, use the **default** keyword followed by the methods you want to use in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to be authenticated, it should be used as a backup method of authentication.

The table below lists the supported NASI authentication methods.

Table 7: AAA Authentication NASI Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.

Keyword	Description
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

NASI Authentication Using Enable Password

Use the **aaa authentication nasi** command with the keyword **enable** to specify the enable password as the authentication method. For example, to specify the enable password as the method of NASI user authentication when no other method list has been defined, use the following command:

```
aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

NASI Authentication Using Line Password

Use the **aaa authentication nasi**command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password. For more information about defining line passwords, refer to the Configuring Line Password Protection.

NASI Authentication Using Local Password

Use the **aaa authentication nasi**command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication information. For example, to specify the local username database as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default local
```

For information about adding users to the local username database, refer to the Establishing Username Authentication.

NASI Authentication Using Group RADIUS

Use the **aaa authentication nasi**command with the **group radius** *method* to specify RADIUS as the NASI authentication method. For example, to specify RADIUS as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

NASI Authentication Using Group TACACS

Use the **aaa authentication nasi**command with the **group tacacs+** *method* keyword to specify TACACS+ as the NASI authentication method. For example, to specify TACACS+ as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

NASI Authentication Using group group-name

Use the **aaa authentication nasi**command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the NASI authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server**command to first define the members of **group nasirad**:

```
aaa group server radius nasirad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *nasirad*.

To specify **group nasirad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Specifying the Amount of Time for Login Input

The **timeout login response**command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **timeout login response** command, you can specify a timeout value from 1 to 300 seconds. To change the login timeout value from the default of 30 seconds, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# timeout login response` *seconds*	Specifies how long the system will wait for login information before timing out.

Enabling Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access the privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication enable default method1 [method2...]`	Enables user ID and password checking for users requesting privileged EXEC level. **Note** All **aaa authentication enable default** requests sent by the router to a RADIUS server include the username "$enab15$." Requests sent to a TACACS+ server will include the username that is entered for login authentication.

The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered. the table below lists the supported enable authentication methods.

Table 8: AAA Authentication Enable Default Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS hosts for authentication. **Note** The RADIUS method does not work on a per-username basis.
group tacacs+	Uses the list of all TACACS+ hosts for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Changing the Text Displayed at the Password Prompt

Use the **aaa authentication password-prompt** command to change the default text that the Cisco IOS XE software displays when prompting a user to enter a password. This command changes the password prompt for the enable password as well as for login passwords that are not supplied by remote security servers. The **no** form of this command returns the password prompt to the following default value:

```
Password:
```

The **aaa authentication password-prompt** command does not change any dialog that is supplied by a remote TACACS+ or RADIUS server.

The aaa authentication password-prompt command works when RADIUS is used as the login method. You will be able to see the password prompt defined in the command shown even when the RADIUS server is unreachable. The aaa authentication password-prompt command does not work with TACACS+. TACACS+ supplies the NAS with the password prompt to display to the users. If the TACACS+ server is reachable, the NAS gets the password prompt from the server and uses that prompt instead of the one defined in the aaa authentication password-prompt command. If the TACACS+ server is not reachable, the password prompt defined in the aaa authentication password-prompt command may be used.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication password-prompt text-string`	Changes the default text displayed when a user is prompted to enter a password.

Preventing an Access Request with a Blank Username from Being Sent to the RADIUS Server

The following configuration steps provide the ability to prevent an Access Request with a blank username from being sent to the RADIUS server. This functionality ensures that unnecessary RADIUS server interaction is avoided, and RADIUS logs are kept short.

Note The **aaa authentication suppress null-username** command is available beginning in Cisco IOS XE Release 2.4.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication suppress null-username**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**aaa new-model** **Example:** `Router(config)# configure terminal`	Enables AAA globally.
Step 4	**aaa authentication suppress null-username** **Example:** `Router(config)# aaa authentication suppress null-username`	Prevents an Access Request with a blank username from being sent to the RADIUS server.

Configuring Message Banners for AAA Authentication

AAA supports the use of configurable, personalized login and failed-login banners. You can configure message banners that will be displayed when a user logs in to the system to be authenticated using AAA and when, for whatever reason, authentication fails.

Configuring a Login Banner

To configure a banner that is displayed when a user logs in (replacing the default message for login), perform the following task:

Before you begin

To create a login banner, you must configure a delimiting character that notifies the system that the following text string must be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string for the banner.

SUMMARY STEPS

1. **aaa new-model** Device(config)# **aaa new-model**
2. Device(config)# **aaa authentication banner** *delimiter string delimiter*
3. Device(config)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**aaa new-model** Device(config)# **aaa new-model**	Enables AAA.
Step 2	Device(config)# **aaa authentication banner** *delimiter string delimiter*	Creates a personalized login banner.
Step 3	Device(config)# **end**	Returns to privileged EXEC mode.

What to do next

After you have configured a login banner, you must complete basic authentication configuration using AAA if you have not already done so. For information about the different types of AAA authentication available, please refer to "Configuring Authentication" in the *Authentication, Authorization, and Accounting Configuration Guide*.

Configuring a Failed-Login Banner

To configure a message that is displayed when a user login fails (replacing the default message for failed login), perform the following task:

Before you begin

To create a failed-login banner, you must configure a delimiting character, which notifies the system that the following text string must be displayed as the banner, and then configure the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

SUMMARY STEPS

1. Device(config)# **aaa new-model**
2. Device(config)# **aaa authentication fail-message** *delimiter string delimiter*
3. Device(config)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **aaa new-model**	Enables AAA.
Step 2	Device(config)# **aaa authentication fail-message** *delimiter string delimiter*	Creates a message to be displayed when a user login fails.
Step 3	Device(config)# **end**	Returns to privileged EXEC mode.

What to do next

After you have configured a failed-login banner, you must complete basic authentication configuration using AAA if you have not already done so. For information about the different types of AAA authentication available, please refer to "Configuring Authentication" in the *Authentication, Authorization, and Accounting Configuration Guide*.

Configuring AAA Packet of Disconnect

Packet of disconnect (POD) terminates connections on the network access server (NAS) when particular session attributes are identified. By using session information obtained from AAA, the POD client residing on a UNIX workstation sends disconnect packets to the POD server running on the network access server. The NAS terminates any inbound user session with one or more matching key attributes. It rejects requests when required fields are missing or when an exact match is not found.

To configure POD, perform the following tasks in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa accounting network default**
2. Router(config)# **aaa accounting delay-start**
3. Router(config)# **aaa pod server server-key**string
4. Router(config)# **radius-server host** *IP address*non-standard

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa accounting network default** **Example:** `start-stop radius`	Enables AAA accounting records.
Step 2	Router(config)# **aaa accounting delay-start**	(Optional) Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 3	Router(config)# **aaa pod server server-key**string	Enables POD reception.
Step 4	Router(config)# **radius-server host** *IP address*non-standard	Declares a RADIUS host that uses a vendor-proprietary version of RADIUS.

Enabling Double Authentication

Depending on the Cisco release, PPP sessions could be authenticated only by using a single authentication method: either PAP or CHAP. Double authentication requires remote users to pass a second stage of authentication (after CHAP or PAP authentication) before gaining network access.

This second ("double") authentication requires a password that is known to the user but *not* stored on the user's remote host. Therefore, the second authentication is specific to a user, not to a host. This provides an additional level of security that will be effective even if information from the remote host is stolen. In addition, this also provides greater flexibility by allowing customized network privileges for each user.

The second stage authentication can use one-time passwords such as token card passwords, which are not supported by CHAP. If one-time passwords are used, a stolen user password is of no use to the perpetrator.

How Double Authentication Works

With double authentication, there are two authentication/authorization stages. These two stages occur after a remote user dials in and a PPP session is initiated.

In the first stage, the user logs in using the remote host name; CHAP (or PAP) authenticates the remote host, and then PPP negotiates with AAA to authorize the remote host. In this process, the network access privileges associated with the remote host are assigned to the user.

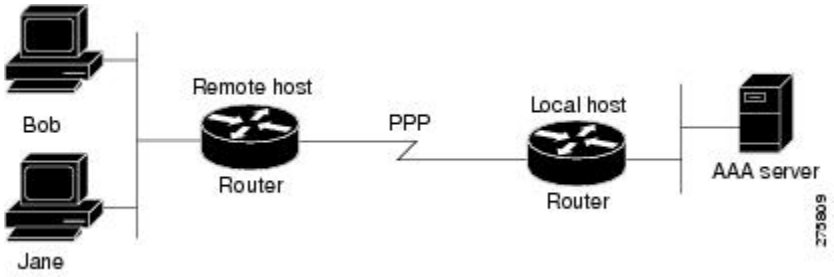
Note We suggest that the network administrator restrict authorization at this first stage to allow only Telnet connections to the local host.

In the second stage, the remote user must Telnet to the network access server to be authenticated. When the remote user logs in, the user must be authenticated with AAA login authentication. The user then must enter the **access-profile** command to be reauthorized using AAA. When this authorization is complete, the user has been double authenticated, and can access the network according to per-user network privileges.

The system administrator determines what network privileges remote users will have after each stage of authentication by configuring appropriate parameters on a security server. To use double authentication, the user must activate it by issuing the **access-profile**command.

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in the figure below. First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per the figure below), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established. Second, if Bob initiates a PPP session and activates double authentication, and then--before Bob's PPP session has expired--another user, Jane, executes the **access-profile** command (or, if Jane Telnets to the network access server and **autocommand access-profile**is executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface--replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 2: Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server



Configuring Double Authentication

To configure double authentication, you must complete the following steps:

1. Enable AAA by using the **aaa-new model**global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.

4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter “Configuring RADIUS”. For more information about TACACS+, refer to the chapter “Configuring TACACS+.”
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile**command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile.

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server.
- If you want remote users to use the interface’s existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration or they can *replace* the existing interface configuration--depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference* .

Accessing the User Profile After Double Authentication

In double authentication, when a remote user establishes a PPP link to the local host using the local host name, the remote host is CHAP (or PAP) authenticated. After CHAP (or PAP) authentication, PPP negotiates with AAA to assign network access privileges associated with the remote host to the user. (We suggest that privileges at this stage be restricted to allow the user to connect to the local host only by establishing a Telnet connection.)

When the user needs to initiate the second phase of double authentication, establishing a Telnet connection to the local host, the user enters a personal username and password (different from the CHAP or PAP username and password). This action causes AAA reauthentication to occur according to the personal username/password. The initial rights associated with the local host, though, are still in place. By using the **access-profile** command, the rights associated with the local host are replaced by or merged with those defined for the user in the user’s profile.

To access the user profile after double authentication, use the following command in EXEC configuration mode:

Command	Purpose
`Router> access-profile [merge | replace] [ignore-sanity-checks]`	Accesses the rights associated for the user after double authentication.

If you configured the **access-profile** command to be executed as an autocommand, it will be executed automatically after the remote user logs in.

Enabling Automated Double Authentication

You can make the double authentication process easier for users by implementing automated double authentication. Automated double authentication provides all of the security benefits of double authentication, but offers a simpler, more user-friendly interface for remote users. With double authentication, a second level of user authentication is achieved when the user Telnets to the network access server or router and enters a username and password. With automated double authentication, the user does not have to Telnet to the network access server; instead the user responds to a dialog box that requests a username and password or personal identification number (PIN). To use the automated double authentication feature, the remote user hosts must be running a companion client application.

Note Automated double authentication, like the existing double authentication feature, is for Multilink PPP ISDN connections only. Automated double authentication cannot be used with other protocols such as X.25 or SLIP.

Automated double authentication is an enhancement to the existing double authentication feature. To configure automated double authentication, you must first configure double authentication by completing the following steps:

1. Enable AAA by using the **aaa-new model**global configuration command.
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the chapter “Configuring Authorization.”
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter “Configuring RADIUS”. For more information about TACACS+, refer to the chapter “Configuring TACACS+.”
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. Configure the **access-profile**command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *CiscoIOS Dial Technologies Command Reference* , Release 12.2.

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration, or *replace* the existing interface configuration--depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference* .

After you have configured double authentication, you are ready to configure the automation enhancement.

Configuring Automated Double Authentication

To configure automated double authentication, use the following commands, starting in global configuration mode.

SUMMARY STEPS

1. Router(config)# **ip trigger-authentication**
2. Do one of the following:
   - Router(config)# **interface bri** *number*
   -
   -
   - Router(config)# **interface serial** *number* **:23**
3. Router(config-if)# **ip trigger-authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **ip trigger-authentication** **Example:** [**timeout** *seconds*] [**port** *number*]	Enables automation of double authentication.
Step 2	Do one of the following: • Router(config)# **interface bri** *number* • •	Selects an ISDN BRI or ISDN PRI interface and enter the interface configuration mode.

	Command or Action	Purpose
	• Router(config)# **interface serial** *number* **:23**	
Step 3	Router(config-if)# **ip trigger-authentication**	Applies automated double authentication to the interface.

Troubleshooting Automated Double Authentication

To troubleshoot automated double authentication, use the following commands in privileged EXEC mode:

SUMMARY STEPS

1. Router# **show ip trigger-authentication**
2. Router# **clear ip trigger-authentication**
3. Router# **debug ip trigger-authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# **show ip trigger-authentication**	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
Step 2	Router# **clear ip trigger-authentication**	Clears the list of remote hosts for which automated double authentication has been attempted. (This clears the table displayed by the **show ip trigger-authentication** command.)
Step 3	Router# **debug ip trigger-authentication**	Displays **debug** output related to automated double authentication.

Configuring the Dynamic Authorization Service for RADIUS CoA

Perform the following steps to enable the device as an authentication, authorization, and accounting (AAA) server for the dynamic authorization service. This service supports the Change of Authorization (CoA) functionality that pushes the policy map in an input and output direction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-addr* | *hostname*} [**server-key** [**0** | **7**] *string*]
6. **domain** {**delimiter** *character* | **stripping** | [**right-to-left**]}
7. **port** *port-num*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Sets up the local AAA server for the dynamic authorization service, which must be enabled to support the CoA functionality to push the policy map in an input and output direction, and enters dynamic authorization local server configuration mode. • In this mode, the RADIUS application commands are configured.
Step 5	**client** {*ip-addr* | *hostname*} [**server-key** [**0** | **7**] *string*] **Example:** `Device(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1`	Configures the IP address or hostname of the AAA server client. • Use the optional **server-key** keyword and *string* argument to configure the server key at the client level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.
Step 6	**domain** {**delimiter** *character* | **stripping** | [**right-to-left**]} **Example:** `Device(config-locsvr-da-radius)# domain stripping right-to-left`	(Optional) Configures username domain options for the RADIUS application. • The **delimiter** keyword specifies the domain delimiter. One of the following options can be specified for the *character* argument: **@**, /, **$**, **%**, \, **#**, or **-**. • The **stripping** keyword compares the incoming username with the names oriented to the left of the **@** domain delimiter. • The **right-to-left** keyword terminates the string at the first delimiter going from right to left.
Step 7	**port** *port-num* **Example:**	Configures the UDP port for CoA requests.

	Command or Action	Purpose
	`Device(config-locsvr-da-radius)# port 3799`	
Step 8	**end** **Example:** `Device(config-locsvr-da-radius)# end`	Returns to privileged EXEC mode.

Configuring a Device to Ignore Bounce and Disable RADIUS CoA Requests

When an authentication port is authenticated with multiple hosts and there is a Change of Authorization (CoA) request for one host to flap on this port or one host session to be terminated on this port, the other hosts on this port are also affected. Thus, an authenticated port with multiple hosts can trigger a DHCP renegotiation from one or more hosts in the case of a flap, or it can administratively shut down the authentication port that is hosting the session for one or more hosts.

Perform the following steps to configure the device to ignore RADIUS server Change of Authorization (CoA) requests in the form of a bounce port command or disable port command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **authentication command bounce-port ignore**
5. **authentication command disable-port ignore**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.
Step 4	**authentication command bounce-port ignore** **Example:**	(Optional) Configures the device to ignore a RADIUS server bounce port command that causes a host to link flap on an

	Command or Action	Purpose
	`Device(config)# authentication command bounce-port ignore`	authentication port, which causes DHCP renegotiation from one or more hosts connected to this port.
Step 5	**authentication command disable-port ignore** **Example:** `Device(config)# authentication command disable-port ignore`	(Optional) Configures the device to ignore a RADIUS server CoA disable port command that administratively shuts down the authentication port that hosts one or more host sessions. • The shutting down of the port causes session termination.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Domain Stripping at the Server Group Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *server-name*
4. **domain-stripping** [**strip-suffix** *word*] [**right-to-left**] [**prefix-delimiter** *word*] [**delimiter** *word*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *server-name* **Example:** `Device(config)# aaa group server radius rad1`	Adds the RADIUS server and enters server group RADIUS configuration mode. • The *server-name* argument specifies the RADIUS server group name.
Step 4	**domain-stripping** [**strip-suffix** *word*] [**right-to-left**] [**prefix-delimiter** *word*] [**delimiter** *word*] **Example:**	Configures domain stripping at the server group level.

	Command or Action	Purpose
	`Device(config-sg-radius)# domain-stripping delimiter username@example.com`	
Step 5	**end** **Example:** `Device(config-sg-radius)# end`	Exits server group RADIUS configuration mode and returns to the privileged EXEC mode.

Non-AAA Authentication Methods

Configuring Line Password Protection

You can This task is used to provide access control on a terminal line by entering the password and establishing password checking.

Note If you configure line password protection and then configure TACACS or extended TACACS, the TACACS username and password take precedence over line passwords. If you have not yet implemented a security policy, we recommend that you use AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
4. **password** *password*
5. **login**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line** [**aux**	**console**	**tty**	**vty**] *line-number* [*ending-line-number*] **Example:**	Enters line configuration mode.

	Command or Action	Purpose
	`Router(config)# line console 0`	
Step 4	**password** *password* **Example:** `Router(config-line)# secret word`	Assigns a password to a terminal or other device on a line. The password checker is case sensitive and can include spaces; for example, the password "Secret" is different from the password "secret," and "two words" is an acceptable password.
Step 5	**login** **Example:** `Router(config-line)# login`	Enables password checking at login. You can disable line password verification by disabling password checking by using the **no** version of this command. **Note** The **login** command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

Establishing Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS
- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and "no escape" situations

To establish username authentication, use the following commands in global configuration mode as needed for your system configuration:

SUMMARY STEPS

1. Do one of the following:
   - Router(config)# **username** *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*]
   -
   -
   - Router(config)# **username** *name* [**access-class** *number*]
2. Router(config)# **username** *name* [**privilege** *level*]
3. Router(config)# **username** *name* [**autocommand** *command*]

4. Router(config)# **username** *name* [**noescape**] [**nohangup**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Do one of the following: • Router(config)# **username** *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*] • • • Router(config)# **username** *name* [**access-class** *number*]	Establishes username authentication with encrypted passwords. or (Optional) Establishes username authentication by access list.
Step 2	Router(config)# **username** *name* [**privilege** *level*]	(Optional) Sets the privilege level for the user.
Step 3	Router(config)# **username** *name* [**autocommand** *command*]	(Optional) Specifies a command to be executed automatically.
Step 4	Router(config)# **username** *name* [**noescape**] [**nohangup**]	(Optional) Sets a "no escape" login environment.

What to do next

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected. The **nohangup** feature does not disconnect after using the autocommand.

⚠

Caution Passwords will be displayed in clear text in your configuration unless you enable the **service password-encryption** command. For more information about the **service password-encryption** command, refer to the *Cisco IOS Security Command Reference*.

Enabling CHAP or PAP Authentication

One of the most common transport protocols used in Internet service providers' (ISPs') dial solutions is the Point-to-Point Protocol (PPP). Traditionally, remote users dial in to an access server to initiate a PPP session. After PPP has been negotiated, remote users are connected to the ISP network and to the Internet.

Because ISPs want only customers to connect to their access servers, remote users are required to authenticate to the access server before they can start up a PPP session. Normally, a remote user authenticates by typing in a username and password when prompted by the access server. Although this is a workable solution, it is difficult to administer and awkward for the remote user.

A better solution is to use the authentication protocols built into PPP. In this case, the remote user dials in to the access server and starts up a minimal subset of PPP with the access server. This does not give the remote user access to the ISP's network--it merely allows the access server to talk to the remote device.

PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and

password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection.

PPP (with or without PAP or CHAP authentication) is also supported in dialout solutions. An access server utilizes a dialout feature when it initiates a call to a remote device and attempts to start up a transport protocol such as PPP.

See the *Cisco IOS XE Dial Technologies Configuration Guide* , Release 2 for more information about CHAP and PAP.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

When the remote device receives the challenge packet, it concatenates the ID, the remote device's password, and the random number, and then encrypts all of it using the remote device's password. The remote device sends the results back to the access server, along with the name associated with the password used in the encryption process.

When the access server receives the response, it uses the name it received to retrieve a password stored in its user database. The retrieved password should be the same password the remote device used in its encryption process. The access server then encrypts the concatenated information with the newly retrieved password--if the result matches the result sent in the response packet, authentication succeeds.

The benefit of using CHAP authentication is that the remote device's password is never transmitted in clear text. This prevents other devices from stealing it and gaining illegal access to the ISP's network.

CHAP transactions occur only at the time a link is established. The access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS XE software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the access server will require authentication from remote devices dialing in to the access server. If the remote device does not support the enabled protocol, the call will be dropped.

To use CHAP or PAP, you must perform the following tasks:

1. Enable PPP encapsulation.
2. Enable CHAP or PAP on the interface.
3. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

Enabling PPP Encapsulation

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP on an interface.

Enabling PAP or CHAP

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp authentication {protocol1 [protocol2...]} [if-needed] {default | list-name} [callin] [one-time]`	Defines the authentication protocols supported and the order in which they are used. In this command, *protocol1*, *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

If you configure **ppp authentication chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using CHAP; likewise, if you configure **ppp authentication pap**, all incoming calls that start a PPP connection will have to be authenticated via PAP. If you configure **ppp authentication chap pap**, the access server will attempt to authenticate all incoming calls that start a PPP session with CHAP. If the remote device does not support CHAP, the access server will try to authenticate the call using PAP. If the remote device does not support either CHAP or PAP, authentication will fail and the call will be dropped. If you configure **ppp authentication pap chap**, the access server will attempt to authenticate all incoming calls that start a PPP session with PAP. If the remote device does not support PAP, the access server will try to authenticate the call using CHAP. If the remote device does not support either protocol, authentication will fail and the call will be dropped. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA--they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication**command with the **if-needed** keyword means that PPP will only authenticate the remote device via PAP or CHAP if they have not yet authenticated during the life of the current call. If the remote device authenticated via a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate via CHAP if **ppp authentication chap if-needed** is configured on the interface.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp**command, you disable PPP on the line.

For information about adding a **username** entry for each remote system from which the local router or access server requires authentication, see the Establishing Username Authentication, on page 38.

Inbound and Outbound Authentication

PPP supports two-way authentication. Normally, when a remote device dials in to an access server, the access server requests that the remote device prove that it is allowed access. This is known as inbound authentication. At the same time, the remote device can also request that the access server prove that it is who it says it is. This is known as outbound authentication. An access server also does outbound authentication when it initiates a call to a remote device.

Enabling Outbound PAP Authentication

To enable outbound PAP authentication, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap sent-username username password password`	Enables outbound PAP authentication.

The access server uses the username and password specified by the **ppp pap sent-username** command to authenticate itself whenever it initiates a call to a remote device or when it has to respond to a remote device's request for outbound authentication.

Refusing PAP Authentication Requests

To refuse PAP authentication from peers requesting it, meaning that PAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap refuse`	Refuses PAP authentication from peers requesting PAP authentication.

If the refuse keyword is not used, the router will not refuse any PAP authentication challenges received from the peer.

Creating a Common CHAP Password

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap password secret`	Enables a router calling a collection of routers to configure a common CHAP secret password.

Refusing CHAP Authentication Requests

To refuse CHAP authentication from peers requesting it, meaning that CHAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap refuse [callin]`	Refuses CHAP authentication from peers requesting CHAP authentication.

If the **callin** keyword is used, the router will refuse to answer CHAP authentication challenges received from the peer, but will still require the peer to answer any CHAP challenges the router sends.

If outbound PAP has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

Delaying CHAP Authentication Until Peer Authenticates

To specify that the router will not authenticate to a peer requesting CHAP authentication until after the peer has authenticated itself to the router, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap wait secret`	Configures the router to delay CHAP authentication until after the peer has authenticated itself to the router.

This command (which is the default) specifies that the router will not authenticate to a peer requesting CHAP authentication until the peer has authenticated itself to the router. The **no ppp chap wait**commandspecifies that the router will respond immediately to an authentication challenge.

Using MS-CHAP

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension of RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco device or access server acting as a network access server.

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.
- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.*x*. This format does not require the authenticator to store a clear or reversibly encrypted password.
- MS-CHAP provides an authenticator-controlled authentication retry mechanism.
- MS-CHAP provides an authenticator-controlled change password mechanism.
- MS-CHAP defines a set of “reason-for failure” codes returned in the Failure packet message field.

Depending on the security protocols you have implemented, PPP authentication using MS-CHAP can be used with or without AAA security services. If you have enabled AAA, PPP authentication using MS-CHAP can

be used in conjunction with both TACACS+ and RADIUS. The table below lists the vendor-specific RADIUS attributes (IETF Attribute 26) that enable RADIUS to support MS-CHAP.

Table 9: Vendor-Specific RADIUS Attributes for MS-CHAP

Vendor-ID Number	Vendor-Type Number	Vendor-Proprietary Attribute	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.

Defining PPP Authentication using MS-CHAP

To define PPP authentication using MS-CHAP, use the following commands in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **encapsulation ppp**
2. Router(config-if)# **ppp authentication ms-chap** [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **encapsulation ppp**	Enables PPP encapsulation.
Step 2	Router(config-if)# **ppp authentication ms-chap** [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]	Defines PPP authentication using MS-CHAP.

What to do next

If you configure **ppp authentication ms-chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using MS-CHAP. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA--they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication**command with the **if-needed** keyword means that PPP will only authenticate the remote device via MS-CHAP if that device has not yet authenticated during the life of the current call. If the remote device

authenticated through a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate through MS-CHAP if **ppp authentication chap if-needed** is configured.

Note If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database. For more information about username authentication, refer to the "Establish Username Authentication" section.

Authentication Examples

RADIUS Authentication Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login radius-login group radius local
aaa authentication ppp radius-ppp if-needed group radius
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
line 3
login authentication radius-login
interface serial 0
ppp authentication radius-ppp
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The aaa authentication login radius-login group radius local command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.
- The aaa authentication ppp radius-ppp if-needed group radius command configures the Cisco IOS XE software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.
- The aaa authorization exec default group radius if-authenticated command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.
- The aaa authorization network default group radius command queries RADIUS for network authorization, address assignment, and other access lists.
- The **login authentication radius-login** command enables the radius-login method list for line 3.
- The **ppp authentication radius-ppp** command enables the radius-ppp method list for serial interface 0.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify it as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

If the user is authenticated using the local database, EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an

autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa authorization command 2 default group tacacs+ if-authenticated
radius-server host 172.16.71.146 auth-port 1645 acct-port 1646
radius-server attribute 44 include-in-access-req
radius-server attribute 8 include-in-access-req
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The aaa authentication login default group radius local command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.
- The aaa authorization exec default group radius local command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The aaa authorization command 2 default group tacacs+ if-authenticated command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.
- The radius-server host 172.16.71.146 auth-port 1645 acct-port 1646 command specifies the IP address of the RADIUS server host, the UDP destination port for authentication requests, and the UDP destination port for accounting requests.
- The radius-server attribute 44 include-in-access-req command sends RADIUS attribute 44 (Acct-Session-ID) in access-request packets.
- The radius-server attribute 8 include-in-access-req command sends RADIUS attribute 8 (Framed-IP-Address) in access-request packets.

TACACS Authentication Examples

The following example shows how to configure TACACS+ as the security protocol to be used for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
interface serial 0
ppp authentication chap pap test
tacacs-server host 192.0.2.3
tacacs-server key goaway
```

The lines in this sample TACACS+ authentication configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **interface** command selects the line.
- The **ppp authentication** command applies the test method list to this line.

- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 192.0.2.3.
- The **tacacs-server key** command defines the shared encryption key to be "goaway."

The following example shows how to configure AAA authentication for PPP:

```
aaa authentication ppp default if-needed group tacacs+ local
```

In this example, the keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP is not necessary and can be skipped. If authentication is needed, the keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa authentication ppp MIS-access if-needed group tacacs+ local
interface serial 0
ppp authentication pap MIS-access
```

In this example, because the list does not apply to any interfaces (unlike the default list, which applies automatically to all interfaces), the administrator must select interfaces to which this authentication scheme should apply by using the **interface** command. The administrator must then apply this method list to those interfaces by using the **ppp authentication** command.

Kerberos Authentication Examples

To specify Kerberos as the login authentication method, use the following command:

```
aaa authentication login default krb5
```

To specify Kerberos authentication for PPP, use the following command:

```
aaa authentication ppp default krb5
```

AAA Scalability Example

The following example shows a general security configuration using AAA with RADIUS as the security protocol. In this example, the network access server is configured to allocate 16 background processes to handle AAA requests for PPP.

```
aaa new-model
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authentication login admins local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa processes 16
line 1 16
 autoselect ppp
```

```
 autoselect during-login
 login authentication admins
 modem dialin
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication pap dialins
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa processes** command allocates 16 background processes to handle AAA requests for PPP.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command allows a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication pap dialins**command applies the "dialins" method list to the specified interfaces.

Example: Configuring Login and Failed-Login Banners for AAA Authentication

The following example shows how to configure a login banner that is displayed when a user logs in to the system, (in this case, the phrase "Unauthorized Access Prohibited"). The asterisk (*) is used as the delimiting character. RADIUS is specified as the default login authentication method.

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication login default group radius
```

This configuration displays the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to configure a failed-login banner that is displayed when a user tries to log in to the system and fails, (in this case, the phrase "Failed login. Try again"). The asterisk (*) is used as the delimiting character. RADIUS is specified as the default login authentication method.

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication fail-message *Failed login. Try again.*
Device(config)# aaa authentication login default group radius
```

This configuration displays the following login and failed-login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

AAA Packet of Disconnect Server Key Example

The following example shows how to configure POD (packet of disconnect), which terminates connections on the network access server (NAS) when particular session attributes are identified.

```
aaa new-model
aaa authentication ppp default radius
aaa accounting network default start-stop radius
aaa accounting delay-start
aaa pod server server-key xyz123
radius-server host 192.0.2.3 non-standard
radius-server key rad123
```

Double Authentication Examples

The examples in this section illustrate possible configurations to be used with double authentication. Your configurations could differ significantly, depending on your network and security requirements.

Note These configuration examples include specific IP addresses and other specific information. This information is for illustration purposes only: your configuration will use different IP addresses, different usernames and passwords, and different authorization statements.

Configuration of the Local Host for AAA with Double Authentication Examples

These two examples show how to configure a local host to use AAA for PPP and login authentication, and for network and EXEC authorization. An example each is shown for RADIUS and for TACACS+.

In both the examples, the first three lines configure AAA with a specific server as the AAA server. The next two lines configure AAA for PPP and login authentication, and the last two lines configure network and EXEC authorization. The last line is necessary only if the **access-profile** command will be executed as an autocommand.

The following example shows device configuration with a RADIUS AAA server:

```
aaa new-model
radius-server host secureserver
radius-server key myradiuskey
aaa authentication ppp default group radius
aaa authentication login default group radius
aaa authorization network default group radius
aaa authorization exec default group radius
```

The following example shows device configuration with a TACACS+ server:

```
aaa new-model
tacacs-server host security
tacacs-server key mytacacskey
aaa authentication ppp default group tacacs+
aaa authentication login default group tacacs+
aaa authorization network default group tacacs+
aaa authorization exec default group tacacs+
```

Configuration of the AAA Server for First-Stage PPP Authentication and Authorization Example

This example shows a configuration on the AAA server. A partial sample AAA configuration is shown for RADIUS.

TACACS+ servers can be configured similarly. (See the Complete Configuration with TACACS Example.)

This example defines authentication/authorization for a remote host named “hostx” that will be authenticated by CHAP in the first stage of double authentication. Note that the ACL AV pair limits the remote host to Telnet connections to the local host. The local host has the IP address 10.0.0.2.

The following example shows a partial AAA server configuration for RADIUS:

```
hostx   Password = “welcome”
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
        cisco-avpair = “lcp:interface-config=ip unnumbered fastethernet 0”,
        cisco-avpair = “ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet”,
        cisco-avpair = “ip:inacl#4=deny icmp any any”,
        cisco-avpair = “ip:route#5=10.0.0.0 255.0.0.0”,
```

```
        cisco-avpair = "ip:route#6=10.10.0.0 255.0.0.0",
        cisco-avpair = "ipx:inacl#3=deny any"
```

Configuration of the AAA Server for Second-Stage Per-User Authentication and Authorization Examples

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication and authorization for a user (Pat) with the username "patuser," who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly. (See the Complete Configuration with TACACS Example.)

Three examples show sample RADIUS AAA configurations that could be used with each of the three forms of the **access-profile** command.

The first example shows a partial sample AAA configuration that works with the default form (no keywords) of the **access-profile** command. Note that only ACL AV pairs are defined. This example also sets up the **access-profile** command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
          cisco-avpair = "ip:inacl#4=deny icmp any any"
```

The second example shows a partial sample AAA configuration that works with the **access-profile merge** form of the **access-profile** command. This example also sets up the **access-profile merge**command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile merge"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any"
          cisco-avpair = "ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.2.0.0 255.255.0.0"
```

The third example shows a partial sample AAA configuration that works with the **access-profile replace** form of the **access-profile** command. This example also sets up the **access-profile replace**command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile replace"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any",
          cisco-avpair = "ip:inacl#4=permit icmp any any",
          cisco-avpair = "ip:route=10.10.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.11.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.12.0.0 255.255.0.0"
```

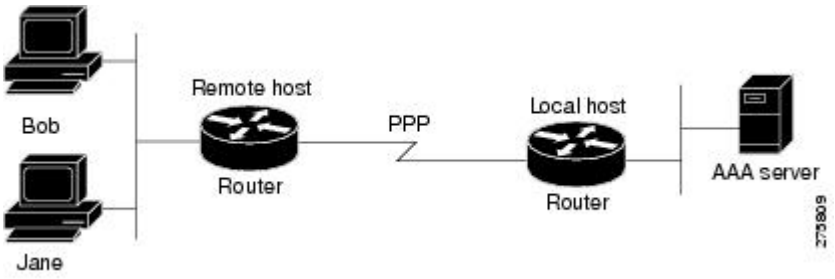
Complete Configuration with TACACS Example

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication). This TACACS+ example contains approximately the same configuration information as shown in the previous RADIUS examples.

This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.” The configurations for these three usernames illustrate different configurations that correspond to the three different forms of the **access-profile** command. The three user configurations also illustrate setting up the autocommand for each form of the **access-profile** command.

The figure below shows the topology. The example that follows the figure shows a TACACS+ configuration file.

Figure 3: Example Topology for Double Authentication



This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.”

```
key = “mytacacskey”
default authorization = permit
#-----------------------------Remote Host (BRI)-------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#-----------------------------------------------------------------------
user = hostx
{
    login = cleartext “welcome”
    chap = cleartext “welcome”
    service = ppp protocol = lcp {
                interface-config=”ip unnumbered fastethernet 0"
    }
    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.
            inacl#3=”permit tcp any 172.21.114.0 0.0.0.255 eq telnet”
            inacl#4=”deny icmp any any”
            route#5=”10.0.0.0 255.0.0.0"
            route#6=”10.10.0.0 255.0.0.0"
    }
    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3=”deny any”
```

```
    }
}
#------------------- "access-profile" default user "only acls" ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#-------------------------------------------------------------------------------
user = pat_default
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when pat_default logs in.
                autocmd = "access-profile"
        }
        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any host 10.0.0.2 eq telnet"
                inacl#4="deny icmp any any"
        }
        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#--------------------- "access-profile merge" user --------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
#
#-------------------------------------------------------------------------------
user = pat_merge
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when pat_merge logs in.
                autocmd = "access-profile merge"
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any any"
                route#2="10.0.0.0 255.255.0.0"
                route#3="10.1.0.0 255.255.0.0"
                route#4="10.2.0.0 255.255.0.0"
```

```
        }
        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#--------------------- "access-profile replace" user ----------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be "invalid" for this context.
#-------------------------------------------------------------------------------
user = pat_replace
{
        login = cleartex
t
"
welcome
"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when pat_replace logs in.
                autocmd = "access-profile replace"
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any any"
                inacl#4="permit icmp any any"
                route#2="10.10.0.0 255.255.0.0"
                route#3="10.11.0.0 255.255.0.0"
                route#4="10.12.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
```

Automated Double Authentication Example

This example shows a complete configuration file with automated double authentication configured. The configuration commands that apply to automated double authentication are preceded by descriptions with a double asterisk (**).

```
Current configuration:
!
version 11.3
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the RADIUS AAA server:
!
aaa authentication login default none
aaa authentication ppp default group radius
! **The following command causes the remote user's authorization profile to be
! downloaded from the AAA server to the router when required:
!
aaa authorization network default group radius
!
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 172.16.2.75
!
!
interface FastEthernet0/0/0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered loopback0
 no ip route-cache
 no ip mroute-cache
!
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 172.16.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
```

```
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines another method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication ms-chap dialins**command selects MS-CHAP as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command allows a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the “admins” method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

Additional References

The following sections provide references related to the Configuring Authentication feature.

Related Documents

Related Topic	Document Title
Authorization	Configuring Authorization in the *Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2.
Accounting	Configuring Accounting in the *Cisco IOS XE Security Configuration Guide: Securing User Service* , Release 2.
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1334	PPP Authentication Protocols
RFC 2433	Microsoft PPP CHAP Extensions
RFC 2903	*Generic AAA Architecture*
RFC 2904	*AAA Authorization Framework*
RFC 2906	*AAA Authorization Requirements*
RFC 2989	*Criteria for Evaluating AAA Protocols for Network Access*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

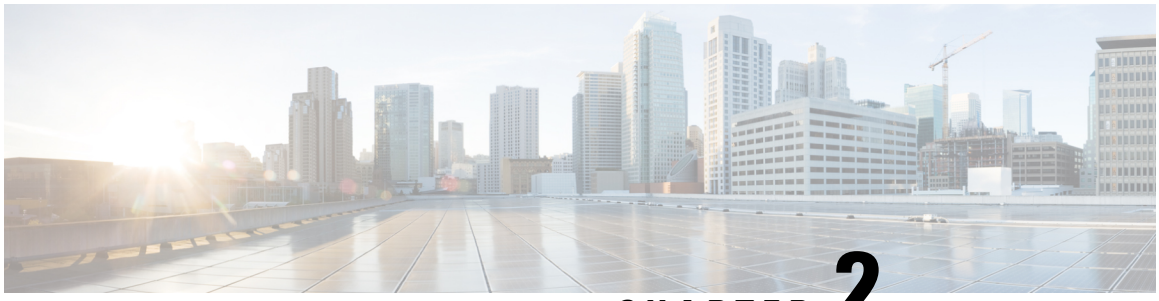
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for Configuring Authentication

Feature Name	Releases	Feature Information
AAA Method Lists Enhancement	Cisco IOS XE Release 2.1	This feature allows you to enable fallback methods for authentication, authorization or accounting. The fallback methods could include trying groups of RADIUS or TACACS+ servers or a local database in some cases. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced or modified: **aaa authentication ppp**.
AAA Per-User Scalability	Cisco IOS XE Release 2.3	The AAA Per-User Scalability feature supports two RADIUS VSAs for **ip vrf** and **ip unnumbered** commands and creates subvirtual access interfaces if specified instead of full VA interface to achieve higher scalability. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Feature Name	Releases	Feature Information
Challege Handshake Authentication Protocol (CHAP)	Cisco IOS XE Release 2.1	PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ppp authentication**, **ppp chap password**, **ppp chap refuse**.
Domain Stripping at the Server Group Level	Cisco IOS XE Release 3.4S	The Domain Stripping feature allows domain stripping to be configured at the server group level. Per-server group configuration overrides the global configuration. The following sections provide information about this feature: • Domain Stripping • Configuring Domain Stripping at the Server Group Level The following command was introduced: **domain-stripping**.
Double Authentication	Cisco IOS XE Release 2.1	In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa authentication**, **aaa authorization**, **access-profile**.
Message Banners for AAA Authentication	Cisco IOS XE Release 2.1	In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced: **aaa authentication banner**.
MS-CHAP Version 1	Cisco IOS XE Release 2.1	Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension of RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ppp authentication**.

Feature Name	Releases	Feature Information
Password Authentication Protocol (PAP)	Cisco IOS XE Release 2.1	PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ppp authentication**, **ppp pap sent-username**, **ppp pap refuse**.
RADIUS—CLI to Prevent Sending of Access Request with a Blank Username	Cisco IOS XE Release 2.4	This authentication feature prevents an Access Request with a blank username from being sent to the RADIUS server. This functionality ensures that unnecessary RADIUS server interaction is avoided, and RADIUS logs are kept short. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced: **aaa authentication suppress null-username**.



CHAPTER 2

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) feature provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. When a policy changes for a user or user group in AAA, administrators can send the RADIUS CoA packets from the AAA server such as a Cisco Secure Access Control Server (ACS) to reinitialize authentication and apply the new policy.

Information About RADIUS Change of Authorization

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. The Cisco software supports the RADIUS CoA request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Use the following per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce
- Security and Password
- Accounting

CoA Requests

CoA requests, as described in RFC 5176, are used in a pushed model to allow for session identification, host reauthentication, and session termination. The model comprises one request (CoA-Request) and two possible response codes:

- CoA acknowledgment (ACK) [CoA-ACK]
- CoA non-acknowledgment (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the device that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the device for a session termination.

The following table shows the IETF attributes that are supported for the RADIUS Change of Authorization (CoA) feature.

Table 11: Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

The following table shows the possible values for the Error-Cause attribute.

Table 12: Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension

Value	Explanation
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request Response code can be used to issue a command to the device. The supported commands are listed in the "CoA Request Commands" section.

The packet format for a CoA Request Response code as defined in RFC 5176 consists of the following fields: Code, Identifier, Length, Authenticator, and Attributes in the Type:Length:Value (TLV) format.

The Attributes field is used to carry Cisco VSAs.

Session Identification

For disconnect and CoA requests targeted at a particular session, the device locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)
- Audit-Session-Id (Cisco vendor-specific attribute (VSA))
- Calling-Station-Id (IETF attribute #31, which contains the host MAC address)

Unless all session identification attributes included in the CoA message match the session, the device returns a Disconnect-NAK or CoA-NAK with the "Invalid Attribute Value" error-code attribute.

Note A CoA NAK message is not sent for all CoA requests with a key mismatch. The message is sent only for the first three requests for a client. After that, all the packets from that client are dropped. When there is a key mismatch, the response authenticator sent with the CoA NAK message is calculated from a dummy key value.

CoA ACK Response Code

If an authorization state is changed successfully, a positive acknowledgment (ACK) is sent. The attributes returned within a CoA ACK can vary based on the CoA Request.

CoA NAK Response Code

A negative acknowledgment (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure.

CoA Request Commands

The commands supported on the device are shown in the table below. All CoA commands must include the session identifier between the device and the CoA client.

Table 13: CoA Request Commands Supported on the Device

Command	Cisco VSA
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the device's response to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1*x*, the device responds by sending an Extensible Authentication Protocol over LAN (EAPoL)-RequestId message to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network. If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism.

CoA Request Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism. This command is carried in a standard CoA-Request message that has the following VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section. If the device cannot locate the session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates the session, it disables the hosting port and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

To ignore the RADIUS server CoA disable port command, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

CoA Request Bounce Port

A RADIUS server CoA bounce port sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer) that does not have a mechanism to detect a change on this authentication port. The CoA bounce port is carried in a standard CoA-Request message that contains the following VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the Session Identification. If the session cannot be located, the device returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device disables the hosting port for a period of 10 seconds, reenables it (port-bounce), and returns a CoA-ACK.

To ignore the RADIUS server CoA bounce port, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

How to Configure RADIUS Change of Authorization

Configuring RADIUS Change of Authorization

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string*
6. **port** *port-number*

7. **auth-type** {**any** | **all** | **session-key**}
8. **ignore session-key**
9. **ignore server-key**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Enters dynamic authorization local server configuration mode and specifies a RADIUS client from which a device accepts Change of Authorization (CoA) and disconnect requests. Configures the device as a AAA server to facilitate interaction with an external policy server.
Step 5	**client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string* **Example:** `Device(config-locsvr-da-radius)# client 10.0.0.1`	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 6	**port** *port-number* **Example:** `Device(config-locsvr-da-radius)# port 3799`	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients. **Note** The default port for packet of disconnect is 1700. Port 3799 is required to interoperate with ACS 5.1.
Step 7	**auth-type** {**any** | **all** | **session-key**} **Example:** `Device(config-locsvr-da-radius)# auth-type all`	Specifies the type of authorization that the device must use for RADIUS clients. The client must match the configured attributes for authorization.
Step 8	**ignore session-key** **Example:** `Device(config-locsvr-da-radius)# ignore session-key`	(Optional) Configures the device to ignore the session key.

	Command or Action	Purpose
Step 9	**ignore server-key** **Example:** `Device(config-locsvr-da-radius)# ignore server-key`	(Optional) Configures the device to ignore the server key.
Step 10	**exit** **Example:** `Device(config-locsvr-da-radius)# exit`	Returns to global configuration mode.

Configuring a Device to Ignore Bounce and Disable RADIUS CoA Requests

When an authentication port is authenticated with multiple hosts and there is a Change of Authorization (CoA) request for one host to flap on this port or one host session to be terminated on this port, the other hosts on this port are also affected. Thus, an authenticated port with multiple hosts can trigger a DHCP renegotiation from one or more hosts in the case of a flap, or it can administratively shut down the authentication port that is hosting the session for one or more hosts.

Perform the following steps to configure the device to ignore RADIUS server Change of Authorization (CoA) requests in the form of a bounce port command or disable port command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **authentication command bounce-port ignore**
5. **authentication command disable-port ignore**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.

	Command or Action	Purpose
Step 4	**authentication command bounce-port ignore** **Example:** `Device(config)# authentication command bounce-port ignore`	(Optional) Configures the device to ignore a RADIUS server bounce port command that causes a host to link flap on an authentication port, which causes DHCP renegotiation from one or more hosts connected to this port.
Step 5	**authentication command disable-port ignore** **Example:** `Device(config)# authentication command disable-port ignore`	(Optional) Configures the device to ignore a RADIUS server CoA disable port command that administratively shuts down the authentication port that hosts one or more host sessions. • The shutting down of the port causes session termination.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the Dynamic Authorization Service for RADIUS CoA

Perform the following steps to enable the device as an authentication, authorization, and accounting (AAA) server for the dynamic authorization service. This service supports the Change of Authorization (CoA) functionality that pushes the policy map in an input and output direction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-addr* | *hostname*} [**server-key** [**0** | **7**] *string*]
6. **domain** {**delimiter** *character* | **stripping** | [**right-to-left**]}
7. **port** *port-num*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Sets up the local AAA server for the dynamic authorization service, which must be enabled to support the CoA functionality to push the policy map in an input and output direction, and enters dynamic authorization local server configuration mode. • In this mode, the RADIUS application commands are configured.
Step 5	**client** {*ip-addr* | *hostname*} [**server-key** [**0** | **7**] *string*] **Example:** `Device(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1`	Configures the IP address or hostname of the AAA server client. • Use the optional **server-key** keyword and *string* argument to configure the server key at the client level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.
Step 6	**domain** {**delimiter** *character* | **stripping** | [**right-to-left**]} **Example:** `Device(config-locsvr-da-radius)# domain stripping right-to-left`	(Optional) Configures username domain options for the RADIUS application. • The **delimiter** keyword specifies the domain delimiter. One of the following options can be specified for the *character* argument: **@**, /, **$**, **%**, \, **#**, or **-**. • The **stripping** keyword compares the incoming username with the names oriented to the left of the **@** domain delimiter. • The **right-to-left** keyword terminates the string at the first delimiter going from right to left.
Step 7	**port** *port-num* **Example:** `Device(config-locsvr-da-radius)# port 3799`	Configures the UDP port for CoA requests.
Step 8	**end** **Example:** `Device(config-locsvr-da-radius)# end`	Returns to privileged EXEC mode.

Monitoring and Troubleshooting RADIUS Change of Authorization

The following commands can be used to monitor and troubleshoot the RADIUS Change of Authorization feature:

Table 14: Monitoring and Troubleshooting RADIUS Change of Authorization

Command	Purpose
debug aaa coa	Displays debug information for CoA processing.
debug aaa pod	Displays debug messages related to packet of disconnect (POD) packets.
debug radius	Displays information associated with RADIUS.
show aaa attributes protocol radius	Displays the mapping between an authentication, authorization, and accounting (AAA) attribute number and the corresponding AAA attribute name.

Configuration Examples for RADIUS Change of Authorization

Example: Configuring RADIUS Change of Authorization

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa server radius dynamic-author
Device(config-locsvr-da-radius)# client 10.0.0.1
Device(config-locsvr-da-radius)# server-key cisco123
Device(config-locsvr-da-radius)# port 3799
Device(config-locsvr-da-radius)# auth-type all
Device(config-locsvr-da-radius)# ignore session-key
Device(config-locsvr-da-radius)# ignore server-key
Device(config-locsvr-da-radius)# end
```

Example: Configuring a Device to Ignore Bounce and Disable a RADIUS Requests

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# authentication command bounce-port ignore
```

```
Device(config)# authentication command disable-port ignore
Device(config)# end
```

Example: Configuring the Dynamic Authorization Service for RADIUS CoA

The following example shows how to configure the device as a authentication, authorization, and accounting (AAA) server to support Change of Authorization (CoA) functionality that pushes the policy map in an input and output direction:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa server radius dynamic-author
Device(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1
Device(config-locsvr-da-radius)# domain delimiter @
Device(config-locsvr-da-radius)# port 3799
Device(config-locsvr-da-radius)# end
```

Additional References for RADIUS Change of Authorization

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Configuring AAA	*Authentication, Authorization, and Accounting Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2903	*Generic AAA Architecture*
RFC 5176	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service(RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

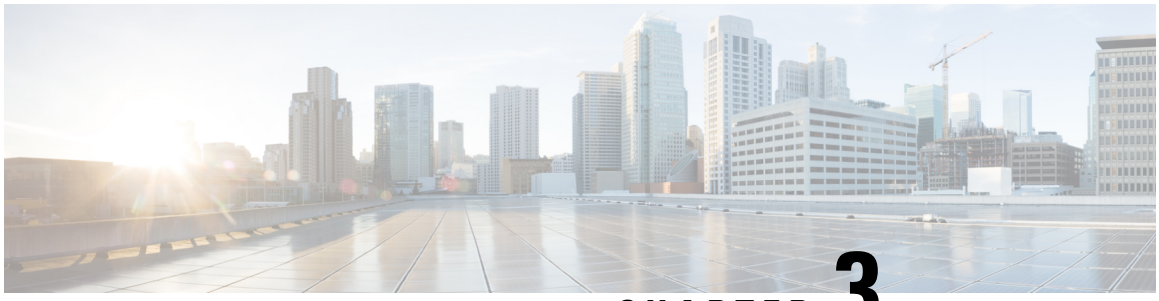
Feature Information for RADIUS Change of Authorization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15: Feature Information for RADIUS Change of Authorization

Feature Name	Releases	Feature Information
RADIUS Change of Authorization		The RADIUS Change of Authorization (CoA) feature provides a mechanism to change the attributes of an AAA session after it is authenticated. When policy changes for a user or user group in AAA, administrators can send the RADIUS CoA packets from the AAA server, such as the Cisco Secure Access Control Server (ACS), to reinitialize authentication and apply the new policy. The following commands were introduced or modified: **aaa server radius dynamic-author**, **authentication command bounce-port ignore**, and **authentication command disable-port ignore**.



CHAPTER 3

Message Banners for AAA Authentication

The Message Banners for AAA authentication feature is used to configure personalized login and failed-login banners for user authentication. The message banners are displayed when a user logs in to the system to be authenticated using authentication, authorization, and accounting (AAA) and when an authentication fails.

Information About Message Banners for AAA Authentication

Login and Failed-Login Banners for AAA Authentication

Login and failed-login banners use a delimiting character that notifies the system of the exact text string that must be displayed as the banner for authorization, authentication, and accounting (AAA) authentication. The delimiting character is repeated at the end of the text string to signify the end of the login or failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string for the banner.

You can display a maximum of 2996 characters in a login or failed-login banner.

How to Configure Message Banners for AAA Authentication

Configuring a Login Banner for AAA Authentication

Perform this task to configure a banner that is displayed when a user logs in (replacing the default message for login). Use the **no aaa authentication banner** command to disable a login banner.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **aaa new-model**
4. **aaa authentication banner** *delimiter-string delimiter*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa authentication banner** *delimiter-string delimiter* **Example:** `Device(config)# aaa authentication banner *Unauthorized Access Prohibited*`	Creates a personalized login banner.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring a Failed-Login Banner for AAA Authentication

Perform this task to configure a failed-login banner that is displayed when a user login fails (replacing the default message for failed login). Use the **no aaa authentication fail-message** command to disable a failed-login banner.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication banner** *delimiter-string delimiter*
5. **aaa authentication fail-message** *delimiter-string delimiter*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enters AAA globally.
Step 4	**aaa authentication banner** *delimiter-string delimiter* **Example:** `Device(config)# aaa authentication banner *Unauthorized Access Prohibited*`	Creates a personalized login banner.
Step 5	**aaa authentication fail-message** *delimiter-string delimiter* **Example:** `Device(config)# aaa authentication fail-message *Failed login. Try again*`	Creates a message to be displayed when a user login fails.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for Message Banners for AAA Authentication

Example: Configuring Login and Failed-Login Banners for AAA Authentication

The following example shows how to configure a login banner that is displayed when a user logs in to the system, (in this case, the phrase “Unauthorized Access Prohibited”). The asterisk (*) is used as the delimiting character. RADIUS is specified as the default login authentication method.

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication login default group radius
```

This configuration displays the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to configure a failed-login banner that is displayed when a user tries to log in to the system and fails, (in this case, the phrase “Failed login. Try again”). The asterisk (*) is used as the delimiting character. RADIUS is specified as the default login authentication method.

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication fail-message *Failed login. Try again.*
Device(config)# aaa authentication login default group radius
```

This configuration displays the following login and failed-login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

Additional References for Message Banners for AAA Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Configuring AAA	*Authentication, Authorization, and Accounting Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

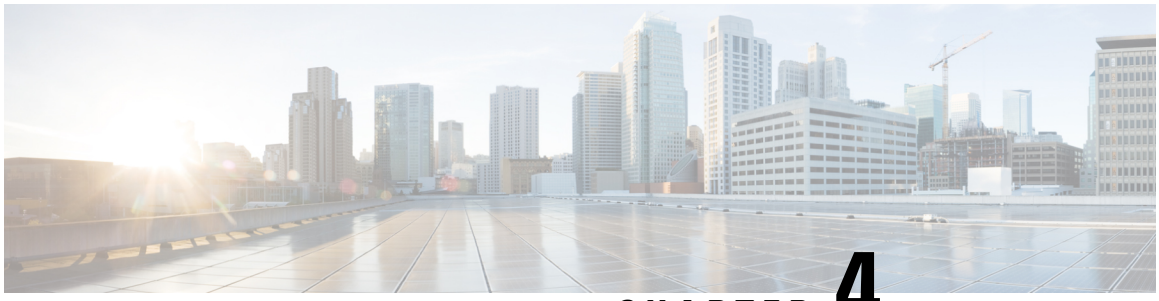
Feature Information for Message Banners for AAA Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16: Feature Information for Message Banners for AAA Authentication

Feature Name	Releases	Feature Information
Message Banners for AAA Authentication		The Message Banners for AAA Authentication feature enables you to configure personalized login and failed-login banners for user authentication. The message banners are displayed when a user logs in to the system to be authenticated using authentication, authorization, and accounting (AAA) and when an authentication fails. The following commands were introduced or modified: **aaa authentication banner**, **aaa authentication fail-message**, and **aaa new-model**.



CHAPTER 4

AAA-Domain Stripping at Server Group Level

The AAA-Domain Stripping at Server Group Level feature allows domain stripping to be configured at the server group level.

Information About AAA-Domain Stripping at Server Group Level

You can remove the domain name from the username received at the global level by using the **radius-server domain-stripping** command. When the **radius-server domain-stripping** command is configured, all the AAA requests with “user@example.com” go to the remote RADIUS server with the reformatted username “user”. The domain name is removed from the request.

Note Domain stripping will not be done in a TACACS configuration.

The AAA Broadcast Accounting feature allows accounting information to be sent to multiple AAA servers at the same time, that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows you to send accounting information to private and public AAA servers. It also provides redundant billing information for voice applications.

You can configure domain stripping at the server group level by using the **domain-stripping** command in server group RADIUS configuration mode. Per-server group configuration overrides the global configuration. If domain stripping is not enabled globally, but it is enabled in a server group, then it is enabled only for that server group. Also, if virtual routing and forwarding (VRF)-specific domain stripping is configured globally and in a server group for a different VRF, domain stripping is enabled in both the VRFs. VRF configurations are taken from server-group configuration mode. If server-group configurations are disabled in global configuration mode but are available in server-group configuration mode, all configurations in server-group configuration mode are applicable.

After the domain stripping and broadcast accounting are configured, you can create separate accounting records as per the configurations.

How to Configure AAA-Domain Stripping at Server Level Group

Configuring Domain Stripping at the Server Group Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *server-name*
5. **domain-stripping** [**strip-suffix** *word*] [**right-to-left**] [**prefix-delimiter** *word*] [**delimiter** *word*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	**aaa group server radius** *server-name* **Example:** `Device(config)# aaa group server radius rad1`	Adds the RADIUS server and enters server group RADIUS configuration mode. • The *server-name* argument specifies the RADIUS server group name.
Step 5	**domain-stripping** [**strip-suffix** *word*] [**right-to-left**] [**prefix-delimiter** *word*] [**delimiter** *word*] **Example:** `Device(config-sg-radius)# domain-stripping delimiter username@example.com`	Configures domain stripping at the server group level.
Step 6	**end** **Example:** `Device(config-sg-radius)# end`	Exits server group RADIUS configuration mode and returns to the privileged EXEC mode.

Configuration Example for AAA-Domain Stripping at Server Group Level

Example: AAA-Domain Stripping at Server Group Level

The following example shows the domain stripping configuration at the server group level:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa group server radius rad1
Device(config-sg-radius)# domain-stripping right-to-left delimiter @$/
Device(config-sg-radius)# end
```

Additional References

The following sections provide references related to the Configuring Authentication feature.

Related Documents

Related Topic	Document Title
Authorization	Configuring Authorization in the *Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2.
Accounting	Configuring Accounting in the *Cisco IOS XE Security Configuration Guide: Securing User Service* , Release 2.
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1334	PPP Authentication Protocols
RFC 2433	Microsoft PPP CHAP Extensions
RFC 2903	*Generic AAA Architecture*
RFC 2904	*AAA Authorization Framework*
RFC 2906	*AAA Authorization Requirements*
RFC 2989	*Criteria for Evaluating AAA Protocols for Network Access*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

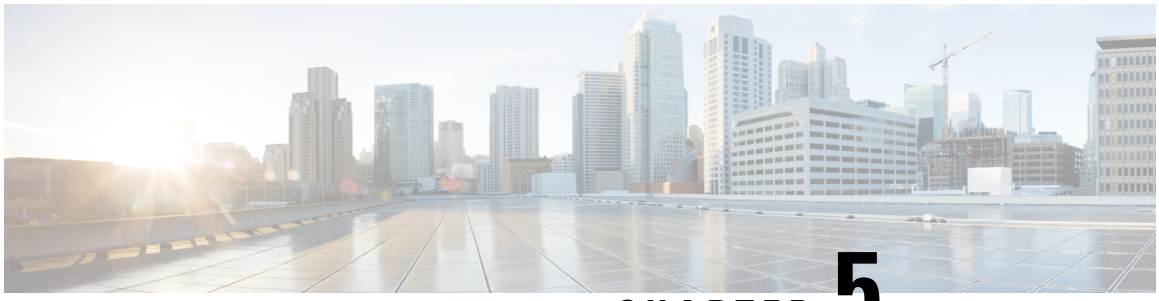
Feature Information for AAA-Domain Stripping at Server Group Level

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17: Feature Information for AAA-Domain Stripping at Server Group Level

Feature Name	Releases	Feature Information
AAA-Domain Stripping at Server Group Level	Cisco IOS XE Release 3.4S	The AAA-Domain Stripping at Server Group Level feature allows domain stripping to be configured at the server group level. The following command was introduced: **domain-stripping**.



CHAPTER 5

AAA Double Authentication Secured by Absolute Timeout

The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. This feature optimizes the connections to the network that are authorized by service providers and increases the security of the overall access to the network by ensuring that no unwanted sessions are connected.

Prerequisites for AAA Double Authentication Secured by Absolute Timeout

- You need access to a Cisco RADIUS or TACACS+ server and should be familiar with configuring RADIUS or TACACS+.
- You should be familiar with configuring authentication, authorization, and accounting (AAA) and enabling AAA automated double authentication.

Restrictions for AAA Double Authentication Secured by Absolute Timeout

- The AAA Double Authentication Secured by Absolute Timeout feature is for PPP connections only. Automated double authentication cannot be used with other protocols, such as X.25 or Serial Line Internet Protocol (SLIP).

- There may be a minimal impact on performance if a TACACS+ server is used. However, there is no performance impact if a RADIUS server is used.

Information About AAA Double Authentication Secured by Absolute Timeout

AAA Double Authentication

Use the AAA double authentication mechanism to pass the first authentication using a host username and password. The second authentication, after the Challenge Handshake Authentication Protocol (CHAP) or the Password Authentication Protocol (PAP) authentication, uses a login username and password. In the first authentication, a PPP session timeout is applied to the virtual access interface if it is configured locally or remotely.

The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. The per-user session timeout, which can be customized, supersedes the generic absolute timeout value. This method works on the same principle as per-user access control lists (ACLs) in double authentication.

How to Apply AAA Double Authentication Secured by Absolute Timeout

Applying AAA Double Authentication Secured by Absolute Timeout

To apply the absolute timeout, you must configure session-timeout in the login user profile as a link control protocol (LCP) per-user attribute. Use the **access-profile** command to enable AAA double authentication. This command is used to apply your per-user authorization attributes to an interface during a PPP session. Before you use the **access-profile** command, you must first reauthorize LCP per-user attributes (for example, Session-Timeout) and then reauthorize Network Control Protocols (NCPs) to apply other necessary criteria, such as ACLs and routes. See the section “Examples for AAA Double Authentication Secured by Absolute Timeout.”

Note The Timeout configuration in a TACACS+ user profile is different from the configuration in a RADIUS user profile. In a RADIUS profile, only one session-timeout is configured, along with the autocommand **access-profile**. The timeout is applied to the EXEC session and to the PPP session respectively. In TACACS+, however, the timeout must be configured under the service types “exec” and “ppp” (LCP) to apply a timeout to the EXEC session and to the PPP session. If the timeout is configured only under the service type “ppp,” the timeout value will not be available during an EXEC authorization, and the timeout will not be applied to the EXEC session.

Configuration Examples for AAA Double Authentication Secured by Absolute Timeout

Example: RADIUS User Profile

The following sample output shows that a RADIUS user profile has been applied and that AAA double authentication has been secured by an absolute timeout:

```
aaapbx2 Password = "password1",
 Service-Type = Framed,
 Framed-Protocol = PPP,
 Session-Timeout = 180,
 Idle-Timeout = 180000,
 cisco-avpair = "ip:inacl#1=permit tcp any any eq telnet"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
broker_default Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
broker_merge Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile merge",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"
broker_replace Password = "password1",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile replace",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"
```

Example: TACACS User Profile

The following sample output shows that a TACACS+ user profile has been applied and that AAA double authentication has been secured by an absolute timeout.

Remote Host Authentication

The following example shows how to allow the remote host to be authenticated by the local host during the first-stage authentication and provides the remote host authorization profile.

```
user = aaapbx2
 chap = cleartext Cisco
 pap = cleartext cisco
 login = cleartext cisco
```

```
 service = ppp protocol = lcp
  idletime = 3000
  timeout = 3
 service = ppp protocol = ip
  inacl#1="permit tcp any any eq telnet"
 service = ppp protocol = ipx
```

Using the access-profile Command Without Any Arguments

Using the **access-profile** command without any arguments causes the removal of any access lists that are found in the old configuration (both per-user and per-interface) and ensures that the new profile contains only access-list definitions.

```
user = broker_default
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile"
! This is the autocommand that executes when broker_default logs in.
  timeout = 6
 service = ppp protocol = lcp
  timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  inacl#1="permit tcp any any"
  inacl#2="permit icmp host 10.0.0.0 any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

Using the access-profile Command with the merge Keyword

The **merge** keyword in the **access-profile** command is used to remove all old access lists, and any attribute-value (AV) pair is allowed to be uploaded and installed. The use of the **merge** keyword will allow for the uploading of any custom static routes, Service Advertisement Protocol (SAP) filters, and other requirements that users may need in their profiles. Configure the **merge** keyword with care because it leaves everything open in terms of conflicting configurations.

```
user = broker_merge
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile merge"
! This is the autocommand that executes when broker_merge logs in.
  timeout = 6
 service = ppp protocol = lcp
 timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  route#1="10.4.0.0 255.0.0.0"
  route#2="10.5.0.0 255.0.0.0"
  route#3="10.6.0.0 255.0.0.0"
```

```
  inacl#5="permit tcp any any"
  inacl#6="permit icmp host 10.60.0.0 any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

Using the access-profile Command with the replace Keyword

If you use the **access-profile** command with the **replace** keyword, any old configurations are removed and a new configuration is installed.

Note When the **access-profile** command is configured, the new configuration is checked for address pools and address-AV pairs. Because addresses cannot be renegotiated at this point, the command will fail to work when it encounters such an address-AV pair.

```
user = broker_replace
 login = cleartext Cisco
 chap = cleartext "cisco"
 service = exec
  autocmd = "access-profile replace"
! This is the autocommand that executes when broker_replace logs in.
  timeout = 6
 service = ppp protocol = lcp
  timeout = 6
 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
  route#1="10.7.0.0 255.0.0.0"
  route#2="10.8.0.0 255.0.0.0"
  route#3="10.9.0.0 255.0.0.0"
  inacl#4="permit tcp any any"
 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

Note The Timeout configuration in a TACACS+ user profile is different from the configuration in a RADIUS user profile. In a RADIUS profile, only one session-timeout is configured, along with the autocommand **access-profile**. The timeout will be applied to the EXEC session and to the PPP session. In the TACACS+ user profile, however, the timeout must be configured under the service types “exec” and “ppp” (LCP) to apply a timeout to the EXEC session and to the PPP session respectively. If the timeout is configured only under the service type “ppp,” the timeout value will not be available during an EXEC authorization, and the timeout will not be applied to the EXEC session.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

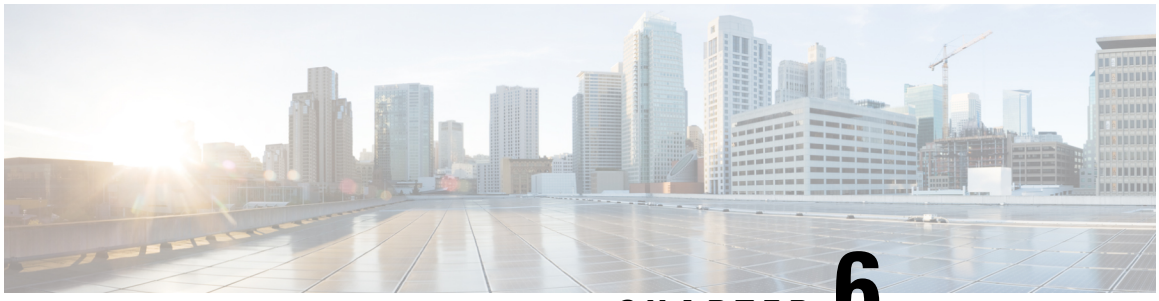
Feature Information for AAA Double Authentication Secured by Absolute Timeout

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 18: Feature Information for AAA Double Authentication Secured by Absolute Timeout

Feature Name	Releases	Feature Information
AAA Double Authentication Secured by Absolute Timeout		The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. This feature optimizes the connection to the network by service providers to only connections that are authorized, and it increases the security of the overall access to the network by ensuring that no unwanted sessions are connected.



CHAPTER 6

Throttling of AAA RADIUS Records

The Throttling of AAA (RADIUS) Records feature supports throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server. This feature allows a user to configure the appropriate throttling rate to avoid network congestion and instability; such as when there is insufficient bandwidth to accommodate a sudden burst of records generated from the router to the RADIUS server.

Information About Throttling of AAA RADIUS Records

Benefits of the Throttling of AAA RADIUS Records Feature

A Network Access Server (NAS), acting as RADIUS client, can generate a burst of accounting or access requests, causing severe network congestion or causing the RADIUS server to become overloaded with a burst of RADIUS traffic. This problem could be compounded when multiple NASs interact with the RADIUS servers.

The following conditions can trigger a sudden burst of RADIUS traffic:

- An interface flap, which in turn brings down all the subscriber sessions and generates accounting requests for each subscriber.
- The High Availability (HA) program generating a START record for every session that survived a switchover, such as the scenario described the preceding bullet.

A large number of generated requests can make the network unstable if there is insufficient bandwidth or if the RADIUS server is slow to respond. Neither the User Datagram Protocol (UDP) transport layer nor the RADIUS protocol has a flow control mechanism. The throttling mechanism provided by this feature provides a solution for these issues.

Throttling Access Requests and Accounting Records

The Throttling of AAA (RADIUS) Records feature introduces a mechanism to control packets (flow control) at the NAS level, which improves the RADIUS server performance.

Because of their specific uses, access requests and accounting records must be treated separately. Access request packets are time sensitive, while accounting record packets are not.

- If a response to an access request is not returned to the client in a timely manner, the protocol or the user will time out, impacting the device transmission rates.
- Accounting records packets are not real-time critical.

When configuring threshold values on the same server, it is important to prioritize threshold values for the handling of the time-sensitive access request packets and to place a lesser threshold value on the accounting records packets.

In some cases, when an Internet Service Provider (ISP) is using separate RADIUS servers for access requests and accounting records, only accounting records throttling may be required.

Summary

- The Throttling of AAA (RADIUS) Records is disabled, by default.
- Throttling functionality can be configured globally or at server group level.

How to Configure Throttling of AAA RADIUS Records

This section describes how to configure throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server for both, global and server groups.

Server-group configurations are used to enable or disable throttling for a particular server group and to specify the threshold value for that server group.

Note Server-group configurations override any configured global configurations.

Throttling Accounting and Access Request Packets Globally

To globally configure the throttling of accounting and access request packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server throttle { accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]}
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server throttle { accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]} **Example:** `Router(config)# radius-server throttle accounting 100 access 200 access-timeout 2`	Configures global throttling for accounting and access request packets. For this example: • The accounting threshold value (the range is 0-65536) is set to 100, and the access threshold value is set to 200. **Note** The default threshold value is 0 (throttling disabled). • The number of timeouts per transaction value (the range is 1-10) is set to 2.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Throttling Accounting and Access Request Packets Per Server Group

The following server-group configuration can be used to enable or disable throttling for a specified server group and to specify the threshold value for that server group.

To configure throttling of server-group accounting and access request packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *server-group-name*
4. **throttle** {[**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *server-group-name* **Example:** `Device(config)# aaa group server radius myservergroup`	Enters server-group configuration mode.
Step 4	**throttle** {[**accounting** *threshold*] [**access** *threshold* [**access-timeout** *number-of-timeouts*]]} **Example:** `Device(config-sg-radius)# throttle accounting 100 access 200 access-timeout 2`	Configures the specified server-group throttling values for accounting and access request packets. For this example: • The accounting threshold value (the range is 0-65536) is set to 100, and the access threshold value is set to 200. **Note** The default threshold value is 0 (throttling disabled). • The number of time-outs per transaction value (the range is 1-10) is set to 2.
Step 5	**exit** **Example:** `Device(config-sg-radius)# exit`	Exits server-group configuration mode.

Configuration Examples for Throttling of AAA RADIUS Records

Throttling Accounting and Access Request Packets Globally Example

The following example shows how to limit the number of accounting requests sent to a server to 100:

```
enable
configure terminal
radius-server throttle accounting 100
```

The following example shows how to limit the number of access requests packets sent to a server to 200 and sets the number of time-outs allowed per transactions to 2:

```
enable
configure terminal
radius-server throttle access 200
radius-server throttle access 200 access-timeout 2
```

The following example shows how to throttle both accounting and access request packets:

```
enable
configure terminal
radius-server throttle accounting 100 access 200
```

Throttling Accounting and Access Request Packets Per Server Group Example

The following example shows how to limit the number of accounting requests sent to server-group-A to 100:

```
enable
configure terminal
aaa group server radius server-group-A
throttle accounting 100
```

The following example shows how to limit the number of access requests packets sent to server-group-A to 200 and sets the number of time-outs allowed per transactions to 2:

```
enable
configure terminal
aaa group server radius server-group-A
throttle access 200 access-timeout 2
```

The following example shows how to throttle both accounting and access request packets for server-group-A:

```
enable
configure terminal
aaa group server radius server-group-A
throttle accounting 100 access 200
```

Additional References

The following sections provide references related to the Throttling of AAA (RADIUS) Records feature.

Related Documents

Related Topic	Document Title
Security features	*Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

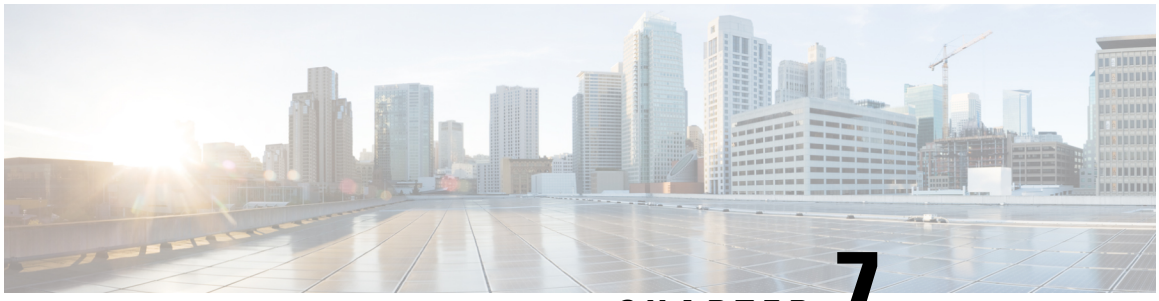
Feature Information for Throttling of AAA RADIUS Records

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 19: Feature Information for Throttling of AAA (RADIUS) Records

Feature Name	Releases	Feature Information
Throttling of AAA (RADIUS) Records	Cisco IOS XE Release 2.1	The Throttling of AAA (RADIUS) Records feature supports throttling of access (authentication and authorization) and accounting records that are sent to the RADIUS server. This feature allows a user to configure the appropriate throttling rate to avoid network congestion and instability; such as when there is insufficient bandwidth to accommodate a sudden burst of records generated from the Cisco IOS XE router to the RADIUS server. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: **radius-server throttle, throttle**



CHAPTER 7

RADIUS Packet of Disconnect

The RADIUS Packet of Disconnect feature is used to terminate a connected voice call.

Prerequisites for RADIUS Packet of Disconnect

Configure AAA as described in the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2.

Restrictions for RADIUS Packet of Disconnect

Proper matching identification information must be communicated by the following:

- Billing server and gateway configuration
- Gateway's original accounting start request
- Server's POD request

Information About RADIUS Packet of Disconnect

The Packet of Disconnect (POD) is a RADIUS access_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect the user after the session has been accepted by the RADIUS access_accept packet.

When the POD is Needed

The POD may be needed in at least two situations:

- Detection of fraudulent use, which cannot be performed before accepting the call. A price structure so complex that the maximum session duration cannot be estimated before accepting the call. This may be the case when certain types of discounts are applied or when multiple users use the same subscription simultaneously.
- To prevent unauthorized servers from disconnecting users, the authorizing agent that issues the POD packet must include three parameters in its packet of disconnect request. For a call to be disconnected, all parameters must match their expected values at the gateway. If the parameters do not match, the gateway discards the packet of disconnect packet and sends a NACK (negative acknowledgement message) to the agent.

POD Parameters

The POD has the following parameters:

- An h323-conf-id vendor-specific attribute (VSA) with the same content as received from the gateway for this call.
- An h323-call-origin VSA with the same content as received from the gateway for the leg of interest.
- A 16-byte MD5 hash value that is carried in the *authentication* field of the POD request.
- Cisco IOS XE software allocates POD code 50 as the code value for the Voice POD Request based on RFC 3576 *Dynamic Authorization Extensions to RADIUS,* which extends RADIUS standards to officially support both a Disconnect Message (DM) and Change-of-Authorization (CoA) that are supported through the POD.

RFC 3576 specifies the following POD codes:

-
  - 40 - Disconnect-Request
  - 41 - Disconnect-ACK
  - 42 - Disconnect-NAK
  - 43 - CoA-Request
  - 44 - CoA-ACK
  - 45 - CoA-NAK

How to Configure the RADIUS Packet of Disconnect

Configuring the RADIUS POD

Use the following tasks to configure the RADIUS POD:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. Router (config)# **aaa pod server** [**port** *port-number*] [**auth-type** {**any**| **all**| **session-key**}] **server-key** [*encryption-type*] *string*
4. Router# **end**
5. Router# **show running-configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	Router (config)# **aaa pod server** [**port** *port-number*] [**auth-type** {**any**| **all**| **session-key**}] **server-key** [*encryption-type*] *string*
	Example:
	`Router(config)# aaa pod server server-key xyz123`
		Enables inbound user sessions to be disconnected when specific session attributes are presented, where:
		• **port** *port-number* --(Optional) The network access server User Datagram Protocol (UDP) port to use for POD requests. Default value is 1700.
		• **auth-type** --(Optional) The type of authorization required for disconnecting sessions.
		  • **any**--Session that matches all of the attributes sent in the POD packet is disconnected. The POD packet may contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key).
		  • **all**--Only a session that matches all four key attributes is disconnected. **All**is the default.
		  • **session-key**--Session with a matching session-key attribute is disconnected. All other attributes are ignored.
		• **server-key**-- Configures the shared-secret text string.
		• *encryption-type* --(Optional) Single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using an encryption algorithm defined by Cisco.
		• *string*-- The shared-secret text string that is shared between the network access server and the client

	Command or Action	Purpose
		workstation. This shared-secret string must be the same on both systems.
Step 4	Router# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Router# **show running-configuration**
	Example:
	`Router# show running-configuration`
	Example:
	`!`
	Example:
	`aaa authentication login h323 group radius`
	Example:
	`aaa authorization exec h323 group radius`
	Example:
	`aaa accounting update newinfo`
	Example:
	`aaa accounting connection h323 start-stop group radius`
	Example:
	`aaa pod server server-key cisco`
	Example:
	`aaa session-id common`
	Example:
	`!`
		Verifies that the gateway is configured correctly in privileged EXEC mode.

Troubleshooting Tips

After you have configured AAA Dead-Server Detection, you should verify your configuration using the **show running-config** command. This verification is especially important if you have used the **no** form of the **radius-server dead-criteria** command. The output of the **show running-config** command must show the same values in the "Dead Criteria Details" field that you configured using the **radius-server dead-criteria** command.

Verifying the RADIUS POD Configuration

To verify the RADIUS POD configuration, use the **show running configuration** privileged EXEC command as shown in the following example:

```
Router# show running-configuration
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting update newinfo
aaa accounting connection h323 start-stop group radius
aaa pod server server-key cisco
aaa session-id common
.
.
.
```

Additional References

The following sections provide references related to the RADIUS Packet of Disconnect feature.

Related Documents

Related Topic	Document Title
AAA	Authentication, Authorization, and Accounting (AAA) section of the *Cisco IOS XE Security Configuration Guide, Securing User Services*, Release 2.
Security commands	*Cisco IOS Security Command Reference*
CLI Configuration	*Cisco IOS XE Configuration Fundamentals Configuration Guide*, Release 2

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial-in User Service*
RFC 3576	*Dynamic Authorization Extensions to RADIUS*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Packet of Disconnect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 20: Feature Information for RADIUS Packet of Disconnect

Feature Name	Releases	Feature Information
RADIUS Packet of Disconnect	Cisco IOS XE Release 2.1	The RADIUS Packet of Disconnect feature is used to terminate a connected voice call. In Cisco IOS XE Release 2.1, this feature was introduced on the on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa pod server, debug aaa pod**

Glossary

AAA --authentication, authorization, and accounting. A framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

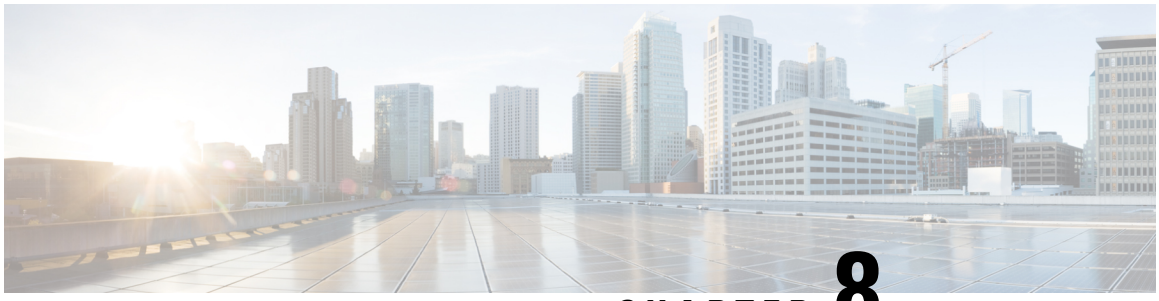
L2TP --Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

PE --Provider Edge. Networking devices that are located on the edge of a service provider network.

RADIUS --Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VPN --Virtual Private Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

VRF --Virtual Route Forwarding. Initially, a router has only one global default routing/forwarding table. VRFs can be viewed as multiple disjoined routing/forwarding tables, where the routes of a user have no correlation with the routes of another user.



CHAPTER 8

AAA Authorization and Authentication Cache

The AAA Authorization and Authentication Cache feature allows you to cache authorization and authentication responses for a configured set of users or service profiles, providing performance improvements and an additional level of network reliability because user and service profiles that are returned from authorization and authentication responses can be queried from multiple sources and need not depend solely on an offload server. This feature also provides a failover mechanism so that if a network RADIUS or TACACS+ server is unable to provide authorization and authentication responses network users and administrators can still access the network.

Prerequisites for Implementing Authorization and Authentication Profile Caching

The following prerequisites apply to implementing authorization and authentication profile caching:

- Understand how you would want to implement profile caching, that is, are profiles being cached to improve network performance or as a failover mechanism if your network authentication and authorization (RADIUS and TACACS+) servers become unavailable.
- RADIUS and TACACS+ server groups must already be configured.

Information About Implementing Authorization and Authentication Profile Caching

Network Performance Optimization Using Authorization and Authentication Profile Caching

RADIUS and TACACS+ clients run on Cisco routers and send authentication requests to a central RADIUS or TACACS+ server that contains all user authentication and network service access information. The router is required to communicate with an offload RADIUS or TACACS+ server to authenticate a given call and then apply a policy or service to that call. Unlike authentication, authorization, and accounting (AAA) accounting, AAA authentication and authorization is a blocking procedure, which means the call setup may not proceed while the call is being authenticated and authorized. Thus, the time required to process the call setup is directly impacted by the time required to process such an authentication or authorization request from the router to the offload RADIUS or TACACS+ server, and back again. Any communication problems in the transmission, offload server utilization, and numerous other factors cause significant degradation in a router's call setup performance due simply to the AAA authentication and authorization step. The problem is further highlighted when multiple AAA authentications and authorizations are needed for a single call or session.

A solution to this problem is to minimize the impact of such authentication requests by caching the authentication and authorization responses for given users on the router, thereby removing the need to send the requests to an offload server again and again. This profile caching adds significant performance improvements to call setup times. Profile caching also provides an additional level of network reliability because user and service profiles that are returned from authentication and authorization responses can be queried from multiple sources and need not depend solely on an offload server.

To take advantage of this performance optimization, you need to configure the authentication method list so that the AAA cache profile is queried first when a user attempts to authenticate to the router. See the Method Lists in Authorization and Authentication Profile Caching section for more information.

Authorization and Authentication Profile Caching as a Failover Mechanism

If, for whatever reason, RADIUS or TACACS+ servers are unable to provide authentication and authorization responses, network users and administrators can be locked out of the network. The profile caching feature allows usernames to be authorized without having to complete the authentication phase. For example, a user by the name of user100@example.com with a password secretpassword1 could be stored in a profile cache using the regular expression ".*@example.com". Another user by the name of user101@example.com with a password of secretpassword2 could also be stored using the same regular expression, and so on. Because the number of users in the ".*@example.com" profile could number in the thousands, it is not feasible to authenticate each user with their personal password. Therefore authentication is disabled and each user simply accesses authorization profiles from a common Access Response stored in cache.

The same reasoning applies in cases where higher end security mechanisms such as Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), or Extensible Authentication Protocol (EAP), which all use an encrypted password between the client and AAA offload server, are used. To allow these unique, secure username and password profiles to retrieve their authorization profiles, authentication is bypassed.

To take advantage of this failover capability, you need to configure the authentication and authorization method list so that the cache server group is queried last when a user attempts to authenticate to the router. See the Method Lists in Authorization and Authentication Profile Caching section for more information.

Method Lists in Authorization and Authentication Profile Caching

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. We support methods such as local (use the local database), none (do nothing), RADIUS server group, or TACACS+ server group. Typically, more than one method can be configured into a method list. Software uses the first listed method to authenticate users. If that method fails to respond, the software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or until all methods defined in the method list are exhausted.

To optimize network performance or provide failover capability using the profile caching feature you simply change the order of the authentication and authorization methods in the method list. To optimize network performance, make sure the cache server group appears first in the method list. For failover capability, the cache server group should appear last in the method list.

Authorization and Authentication Profile Caching Guidelines

Because the number of usernames and profiles that can request to be authenticated or authorized at a given router on a given point of presence (POP) can be quite extensive, it would not be feasible to cache all of them. Therefore, only usernames and profiles that are commonly used or that share a common authentication and authorization response should be configured to use caching. Commonly used usernames such as aolip and aolnet, which are used for America Online (AOL) calls, or preauthentication dialed number identification service (DNIS) numbers used to connect Public Switched Telephone Network (PSTN) calls to a network attached storage device, along with domain-based service profiles, are all examples of usernames and profiles that can benefit from authentication and authorization caching.

General Configuration Procedure for Implementing Authorization and Authentication Profile Caching

To implement authorization and authentication profile caching, you would complete the following procedure:

1. Create cache profile groups and define the rules for what information is cached in each group.

Entries that match based on exact username, regular expressions, or specify that all authentication and authorization requests can be cached.

1. Update existing server groups to reference newly defined cache groups.
2. Update authentication or authorization method lists to use the cached information to optimize network performance or provide a failover mechanism.

How to Implement Authorization and Authentication Profile Caching

Creating Cache Profile Groups and Defining Caching Rules

Perform this task to create a cache profile group, define the rules for what information is cached in that group, and verify and manage cache profile entries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa cache profile** *group-name*
5. **profile** *name* [**no-auth**]
6. Repeat Step 5 for each username you want to add to the profile group in Step 4.
7. **regexp** *matchexpression* {**any**| **only**}[**no-auth**]
8. Repeat Step 7 for each regular expression you want to add to the cache profile group defined in Step 4.
9. **all** [**no-auth**]
10. **end**
11. **show aaa cache group** *name*
12. **clear aaa cache group** *name* {**profile** *name*| **all**}
13. **debug aaa cache group**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa cache profile** *group-name* **Example:**	Defines an authentication and authorization cache profile server group and enters profile map configuration mode.

	Command or Action	Purpose
	`Router(config)# aaa cache profile networkusers@companyname`	
Step 5	**profile** *name* [**no-auth**] **Example:** `Router(config-profile-map# profile networkuser1 no-auth`	Creates an individual authentication and authorization cache profile based on a username match. • The *name* argument must be an exact match to a username being queried by an authentication or authorization service request. • Use the **no-auth** keyword to bypass authentication for this user.
Step 6	Repeat Step 5 for each username you want to add to the profile group in Step 4.	--
Step 7	**regexp** *matchexpression* {**any**| **only**}[**no-auth**] **Example:** `Router(config-profile-map)# regexp .*@example.com any no-auth`	(Optional) Creates an entry in a cache profile group that matches based on a regular expression. • If you use the **any** keyword, all unique usernames matching the regular expression are saved. • If you use the **only** keyword, only one profile entry is cached for all usernames matching the regular expression. • Use the **no-auth** keyword to bypass authentication for this user or set of users. • Because the number of entries in a regular expression cache profile group could be in the thousands, and validating each request against a regular expression can be time consuming, we do not recommend using regular expression entries in cache profile groups.
Step 8	Repeat Step 7 for each regular expression you want to add to the cache profile group defined in Step 4.	--
Step 9	**all** [**no-auth**] **Example:** `Router(config-profile-map)# all no-auth`	(Optional) Specifies that all authentication and authorization requests are cached. • Use the **all** command for specific service authorization requests, but it should be avoided when dealing with authentication requests.
Step 10	**end** **Example:** `Router(config-profile-map)# end`	Returns to privileged EXEC mode.
Step 11	**show aaa cache group** *name* **Example:**	(Optional) Displays all cache entries for a specified group.

	Command or Action	Purpose
	`Router# show aaa cache group networkusers@companyname`	
Step 12	**clear aaa cache group** *name* {**profile** *name*| **all**} **Example:** `Router# clear aaa cache group networkusers@companyname profile networkuser1`	(Optional) Clears an individual entry or all entries in the cache.
Step 13	**debug aaa cache group** **Example:** `Router# debug aaa cache group`	(Optional) Displays debug information about cached entries.

Defining RADIUS and TACACS Server Groups That Use Cache Profile Group Information

Perform this task to define how RADIUS and TACACS+ server groups use the information stored in each cache profile group.

Before you begin

RADIUS and TACACS+ server groups must be created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *group-name* or**aaa group server tacacs+** *group-name*
5. **cache authorization profile** *name*
6. **cache authentication profile** *name*
7. **cache expiry** *hours* {**enforce failover**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa group server radius** *group-name* or**aaa group server tacacs+** *group-name* **Example:** `Router(config)# aaa group server radius networkusers@companyname`	Enters RADIUS server group configuration mode. • To enter TACACS+ server group configuration mode, use the **aaa group server tacacs+** *group-name* command.
Step 5	**cache authorization profile** *name* **Example:** `Router(config-sg-radius)# cache authorization profile networkusers@companyname`	Activates the authorization caching rules in the profile networkusers for this RADIUS or TACACS+ server group. • The *name* argument in this command is a AAA cache profile group name.
Step 6	**cache authentication profile** *name* **Example:** `Router(config-sq-radius)# cache authentication profile networkusers@companyname`	Activates the authentication caching rules in the profile networkusers for this RADIUS or TACACS+ server group.
Step 7	**cache expiry** *hours* {**enforce failover**} **Example:** `Router(config-sq-radius)# cache expiry 240 failover`	(Optional) Sets the amount of time before a cache profile entry expires (becomes stale). • Use the **enforce** keyword to specify that once a cache profile entry expires it is not used again. • Use the **failover** keyword to specify that an expired cache profile entry can be used if all other methods to authenticate and authorize the user fail.
Step 8	**end** **Example:** `Router(config-sg-radius)# end`	Returns to privileged EXEC mode.

Updating Authorization and Authentication Method Lists to Specify How Cache Information is Used

Perform this task to update authorization and authentication method lists to use the authorization and authentication cache information.

Before you begin

Method lists must already be defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access**| **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
5. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
6. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization** {**network**	**exec**	**commands** *level*	**reverse-access**| **configuration**} {**default**	*list-name*} [*method1* [*method2...*]] **Example:** `Router(config)# aaa authorization network default cache networkusers@companyname group networkusers@companyname`	Enables AAA authorization and creates method lists, which define the authorization methods used when a user accesses a specified function.
Step 5	**aaa authentication ppp** {**default**	*list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp default cache networkusers@companyname group networkusers@companyname`	Specifies one or more authentication methods for use on serial interfaces that are running PPP.

	Command or Action	Purpose
Step 6	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication login default cache adminusers group adminusers`	Sets the authentication at login.
Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for Implementing Authorization and Authentication Profile Caching

Implementing Authorization and Authentication Profile Caching for Network Optimization Example

The following configuration example shows how to:

- Define a cache profile group adminusers that contains all administrator names on the network and sets it as the default list that is used for all login and exec sessions.
- Activate the new caching rules for a RADIUS server group.
- Add the new cache profile group in the authentication and authorization method list and change the method order so that the cache profile group is queried first.

```
configure terminal

 aaa new-model

 ! Define aaa cache profile groups and the rules for what information is saved to cache.

 aaa cache profile admin_users

 profile adminuser1

 profile adminuser2

 profile adminuser3

 profile adminuser4
```

```
 profile adminuser5

 exit

 ! Define server groups that use the cache information in each profile group.

 aaa group server radius admins@companyname.com

 cache authorization profile admin_users

 cache authentication profile admin_users

 ! Update authentication and authorization method lists to specify how profile groups and
server groups are used.

 aaa authentication login default cache admins@companyname.com group admins@companyname.com

 aaa authorization exec default cache admins@companyname.com group admins@companyname.com

 end
```

Implementing Authorization and Authentication Profile Caching as a Failover Mechanism Example

The following configuration example shows how to:

- Create a cache profile group admin_users that contains all of the administrators on the network so that if the RADIUS or TACACS+ server should become unavailable the administrators can still access the network.
- Create a cache profile group abc_users that contains all of the ABC company users on the network so that if the RADIUS or TACACS+ server should become unavailable these users will be authorized to use the network.
- Activate the new caching rules for each profile group on a RADIUS server.
- Add the new cache profile group in the authentication and authorization method list and change the method order so that the cache profile group is queried last.

```
configure terminal

 aaa new-model

 ! Define aaa cache profile groups and the rules for what information is saved to cache.

 aaa cache profile admin_users

 profile admin1
```

```
profile admin2
profile admin3
exit
aaa cache profile abcusers
profile .*@example.com only no-auth
exit
! Define server groups that use the cache information in each cache profile group.
aaa group server tacacs+ admins@companyname.com
server 10.1.1.1
server 10.20.1.1
cache authentication profile admin_users
cache authorization profile admin_users
exit
aaa group server radius abcusers@example.com
server 172.16.1.1
server 172.20.1.1
cache authentication profile abcusers
cache authorization profile abcusers
exit
! Update authentication and authorization method lists to specify how cache is used.
aaa authentication login default cache admins@companyname.com group admins@companyname.com
aaa authorization exec default cache admins@companyname.com group admins@companyname.com
aaa authentication ppp default group abcusers@example.com cache abcusers@example.com
aaa authorization network default group abcusers@example.com cache abcusers@example.com
```

```
end
```

Additional References for RADIUS Change of Authorization

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Configuring AAA	*Authentication, Authorization, and Accounting Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2903	*Generic AAA Architecture*
RFC 5176	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service(RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

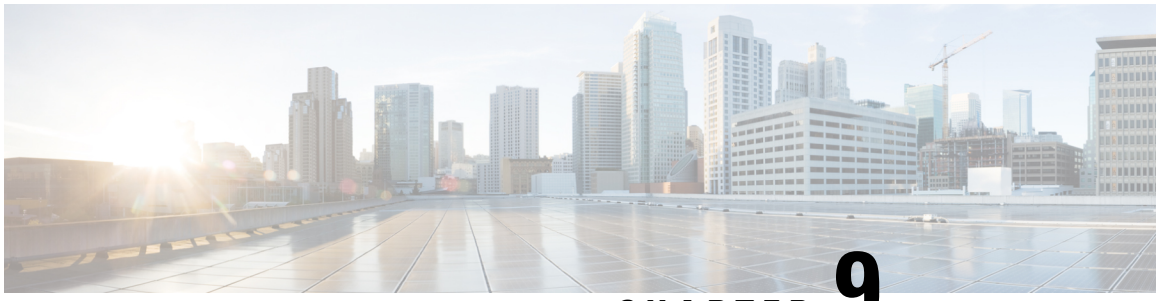
Feature Information for Implementing Authorization and Authentication Profile Caching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 21: Feature Information for Implementing Authentication and Authorization Profile Caching

Feature Name	Release	Feature Information
AAA Authorization and Authentication Cache	Cisco IOS XE Release 2.3	This feature optimizes network performance and provides a failover mechanism in the event a network RADIUS or TACACS+ server becomes unavailable for any reason. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa authentication login**, **aaa authentication ppp**, **aaa authorization**, **aaa cache profile**, **all (profile map configuration)**, **cache authentication profile (server group configuration)**, **cache authorization profile (server group configuration)**, **cache expiry (server group configuration)**, **clear aaa cache group**, **debug aaa cache group**, **profile (profile map configuration)**, **regexp (profile map configuration)**, **show aaa cache group**.



CHAPTER 9

Configuring Authorization

AAA authorization enables you to limit the services available to a user. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user will be granted access to a requested service only if the information in the user profile allows it.

AAA Authorization Prerequisites

Before configuring authorization using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server.
- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly. For more information about AAA authentication, refer to the "Configuring Authentication" module.
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the chapter "Configuring RADIUS". For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the "Configuring TACACS+" module.
- Define the rights associated with specific users by using the **username** command if you are issuing local authorization. For more information about the **username** command, refer to the *Cisco IOS Security Command Reference* .

Information About Configuring Authorization

Named Method Lists for Authorization

Method lists for authorization define the ways that authorization will be performed and the sequence in which these methods will be performed. A method list is simply a named list describing the authorization methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS XE software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS XE software selects the next method listed in the list. This process continues until there is successful communication with a listed authorization method, or all methods defined are exhausted.

Note The Cisco IOS XE software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle--meaning that the security server or local username database responds by denying the user services--the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the authorization type requested:

- Commands--Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- EXEC--Applies to the attributes associated with a user EXEC terminal session.
- Network--Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- Reverse Access--Applies to reverse Telnet sessions.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named “default”). If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, local authorization takes place by default.

AAA Authorization Methods

AAA supports five different methods of authorization:

- TACACS+—The network access server exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user.

- If-Authenticated—The user is allowed to access the requested function provided the user has been authenticated successfully.
- None—The network access server does not request authorization information; authorization is not performed over this line/interface.
- Local—The router or access server consults its local database, as defined by the **username** command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database.
- RADIUS—The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

Note With CSCuc32663, passwords and authorization logs are masked before being sent to the TACACS+, LDAP, or RADIUS security servers. Use the **aaa authorization commands visible-keys** command to send unmasked information to the TACACS+, LDAP, or RADIUS security servers.

Authorization Methods

To have the network access server request authorization information via a TACACS+ security server, use the **aaa authorization** command with the **group tacacs+** *method* keyword. For more specific information about configuring authorization using a TACACS+ security server, refer to the chapter "Configuring TACACS+." For an example of how to enable a TACACS+ server to authorize the use of network services, including PPP and ARA, see the TACACS Authorization Examples.

To allow users to have access to the functions they request as long as they have been authenticated, use the **aaa authorization** command with the **if-authenticated** *method* keyword. If you select this method, all requested functions are automatically granted to authenticated users.

There may be times when you do not want to run authorization from a particular interface or line. To stop authorization activities on designated lines or interfaces, use the **none** *method* keyword. If you select this method, authorization is disabled for all actions.

To select local authorization, which means that the router or access server consults its local user database to determine the functions a user is permitted to use, use the **aaa authorization** command with the **local** *method* keyword. The functions associated with local authorization are defined by using the **username** global configuration command. For a list of permitted functions, refer to the chapter "Configuring Authentication."

To have the network access server request authorization via a RADIUS security server, use the **radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the Configuring RADIUS chapter.

To have the network access server request authorization via a RADIUS security server, use the **aaa authorization** command with the **group radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the chapter Configuring RADIUS. For an example of how to enable a RADIUS server to authorize services, see the RADIUS Authorization Example.

Note Authorization method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for authorization applies.

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This means you can specify either R1 and T1 in the method list or R2 and T2 in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, authorization--the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the chapter Configuring RADIUS or the chapter Configuring TACACS+.

AAA Authorization Types

Cisco IOS XE software supports five different types of authorization:

- Commands--Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- EXEC--Applies to the attributes associated with a user EXEC terminal session.
- Network--Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- Reverse Access--Applies to reverse Telnet sessions.
- Configuration--Applies to downloading configurations from the AAA server.
- IP Mobile--Applies to authorization for IP mobile services.

Authorization Types

Named authorization method lists are specific to the indicated type of authorization.

To create a method list to enable authorization that applies specific security policies on a per-user basis, use the auth-proxy keyword. For detailed information on the authentication proxy feature, refer to the chapter "Configuring Authentication Proxy" in the "Traffic Filtering and Firewalls" part of this book.

To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARAP), use the **network** keyword.

To create a method list to enable authorization to determine if a user is allowed to run an EXEC shell, use the **exec** keyword.

To create a method list to enable authorization for specific, individual EXEC commands associated with a specific privilege level, use the **commands**keyword. (This allows you to authorize all commands associated with a specified command level from 0 to 15.)

To create a method list to enable authorization for reverse Telnet functions, use the **reverse-access** keyword.

For information about the types of authorization supported by the Cisco IOS XE software, refer to the AAA Authorization Types.

Authorization Attribute-Value Pairs

RADIUS and TACACS+ authorization both define specific rights for users by processing attributes, which are stored in a database on the security server. For both RADIUS and TACACS+, attributes are defined on the security server, associated with the user, and sent to the network access server where they are applied to the user's connection.

For a list of supported RADIUS attributes, refer to the "RADIUS Attributes Overview and RADIUS IETF Attributes" chapter. For a list of supported TACACS+ AV pairs, refer to the "Configuring TACACS+" chapter.

How to Configure Authorization

For authorization configuration examples using the commands in this chapter, refer to the Authorization Configuration Examples.

Configuring AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2...*]]
2. Do one of the following:
   - Router(config)# **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
   -
   -
   - Router(config)# **interface** *interface-type interface-number*
3. Do one of the following:

- Router(config-line)# **authorization**{**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
-
-
- Router(config-line)# **ppp authorization**{**default** | *list-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2...*]]	Creates an authorization method list for a particular authorization type and enable authorization.
Step 2	Do one of the following: • Router(config)# **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] • • • Router(config)# **interface** *interface-type interface-number*	Enters the line configuration mode for the lines to which you want to apply the authorization method list. Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.
Step 3	Do one of the following: • Router(config-line)# **authorization**{**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*} • • • Router(config-line)# **ppp authorization**{**default** | *list-name*}	Applies the authorization list to a line or set of lines. Alternately, applies the authorization list to an interface or set of interfaces.

Disabling Authorization for Global Configuration Commands

The **aaa authorization** command with the keyword **commands** attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level. Because there are configuration commands that are identical to some EXEC-level commands, there can be some confusion in the authorization process. Using **no aaa authorization config-commands** stops the network access server from attempting configuration command authorization.

To disable AAA authorization for all global configuration commands, use the following command in global configuration mode:

Command	Purpose
`Device(config)# no aaa authorization config-commands`	Disables authorization for all global configuration commands.

To disable AAA authorization on the console, use the following command in global configuration mode:

Note AAA authorization is disabled on the console by default. If AAA authorization is enabled on the console, disable it by configuring the **no aaa authorization console** command during the AAA configuration stage. AAA should be disabled on the console for user authentication.

Command	Purpose
`Device(config)# no aaa authorization console`	Disables authorization on the console.

Configuring Authorization for Reverse Telnet

Telnet is a standard terminal emulation protocol used for remote terminal connection. Normally, you log in to a network access server (typically through a dialup connection) and then use Telnet to access other network devices from that network access server. There are times, however, when it is necessary to establish a reverse Telnet session. In reverse Telnet sessions, the Telnet connection is established in the opposite direction--from inside a network to a network access server on the network periphery to gain access to modems or other devices connected to that network access server. Reverse Telnet is used to provide users with dialout capability by allowing them to Telnet to modem ports attached to a network access server.

It is important to control access to ports accessible through reverse Telnet. Failure to do so could, for example, allow unauthorized users free access to modems where they can trap and divert incoming calls or make outgoing calls to unauthorized destinations.

Authentication during reverse Telnet is performed through the standard AAA login procedure for Telnet. Typically the user has to provide a username and password to establish either a Telnet or reverse Telnet session. Reverse Telnet authorization provides an additional (optional) level of security by requiring authorization in addition to authentication. When enabled, reverse Telnet authorization can use RADIUS or TACACS+ to authorize whether or not this user is allowed reverse Telnet access to specific asynchronous ports, after the user successfully authenticates through the standard Telnet login procedure.

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.
- An alternative method (other than access lists) to manage reverse Telnet authorization.

To configure a network access server to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse Telnet session, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authorization reverse-access` *method1* `[`*method2* `...]`	Configures the network access server to request authorization information before allowing a user to establish a reverse Telnet session.

This feature enables the network access server to request reverse Telnet authorization information from the security server, whether RADIUS or TACACS+. You must configure the specific reverse Telnet privileges for the user on the security server itself.

Authorization Configuration Examples

TACACS Authorization Examples

The following examples show how to use a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or an error occurs during the authorization process, the fallback method (none) is to grant all authorization requests:

```
aaa authorization network default group tacacs+ none
```

The following example shows how to allow network authorization using TACACS+:

```
aaa authorization network default group tacacs+
```

The following example shows how to provide the same authorization, but it also creates address pools called "*mci*" and "*att*" :

```
aaa authorization network default group tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext "some password"
    service = ppp protocol = ip {
        addr-pool=mci
    }
}
user = att_customer1 {
    login = cleartext "some other password"
    service = ppp protocol = ip {
        addr-pool=att
     }
```

RADIUS Authorization Example

The following example shows how to configure the router to authorize using RADIUS:

```
aaa new-model
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
radius-server host ip
radius-server key
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group radius if-authenticated** command configures the network access server to contact the RADIUS server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the RADIUS server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.

- The **aaa authorization network default group radius** command configures network authorization via RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Note Because no fallback method is specified in this example, authorization will fail if, for any reason, there is no response from the RADIUS server.

Reverse Telnet Authorization Examples

The following examples show how to cause the network access server to request authorization information from a TACACS+ security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization reverse-access default group tacacs+
!
tacacs-server host 172.31.255.0
tacacs-server timeout 90
tacacs-server key goaway
```

The lines in this sample TACACS+ reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group tacacs+** command specifies TACACS+ as the default method for user authentication during login.
- The **aaa authorization reverse-access default group tacacs+** command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.
- The **tacacs-server host** command identifies the TACACS+ server.
- The **tacacs-server timeout**command sets the interval of time that the network access server waits for the TACACS+ server to reply.
- The **tacacs-server key** command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

The following example shows how to configure a generic TACACS+ server to grant a user, pat, reverse Telnet access to port tty2 on the network access server named “maple” and to port tty5 on the network access server named “oak”:

```
user = pat
  login = cleartext lab
  service = raccess {
    port#1 = maple/tty2
    port#2 = oak/tty5
```

Note In this example, “maple” and “oak” are the configured host names of network access servers, not DNS names or alias.

The following example shows how to configure the TACACS+ server (CiscoSecure) to grant a user named pat reverse Telnet access:

```
user = pat
profile_id = 90
profile_cycle = 1
member = Tacacs_Users
service=shell {
default cmd=permit
}
service=raccess {
allow "c2511e0" "tty1" ".*"
refuse ".*" ".*" ".*"
password = clear "goaway"
```

Note CiscoSecure only supports reverse Telnet using the command line interface in versions 2.1(*x*) through version 2.2(1).

An empty “service=raccess {}” clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no “service=raccess” clause exists, the user is denied access to any port for reverse Telnet.

For more information about configuring TACACS+, refer to the “Configuring TACACS” chapter. For more information about configuring CiscoSecure, refer to the *CiscoSecure Access Control Server User Guide* , version 2.1(2) or greater.

The following example shows how to cause the network access server to request authorization from a RADIUS security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group radius
aaa authorization reverse-access default group radius
!
radius-server host 172.31.255.0
radius-server key go away
auth-port 1645 acct-port 1646
```

The lines in this sample RADIUS reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group radius** command specifies RADIUS as the default method for user authentication during login.
- The **aaa authorization reverse-access default group radius** command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.
- The **radius-server host** command identifies the RADIUS server.
- The **radius-server key** command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

The following example shows how to send a request to the RADIUS server to grant a user named “pat” reverse Telnet access at port tty2 on the network access server named “maple”:

```
Username = “pat”
Password = “goaway”
User-Service-Type = Shell-User
cisco-avpair = “raccess:port#1=maple/tty2”
```

The syntax "raccess:port=any/any" permits a user to have unconditional access to network access server ports for reverse Telnet. If no "raccess:port={*nasname* }/{*tty number* }" clause exists in the user profile, the user is denied access to reverse Telnet on all ports.

For more information about configuring RADIUS, refer to the chapter “Configuring RADIUS.”

Additional References

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference
IPsec	IPsec Virtual Tunnel Interface feature document

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

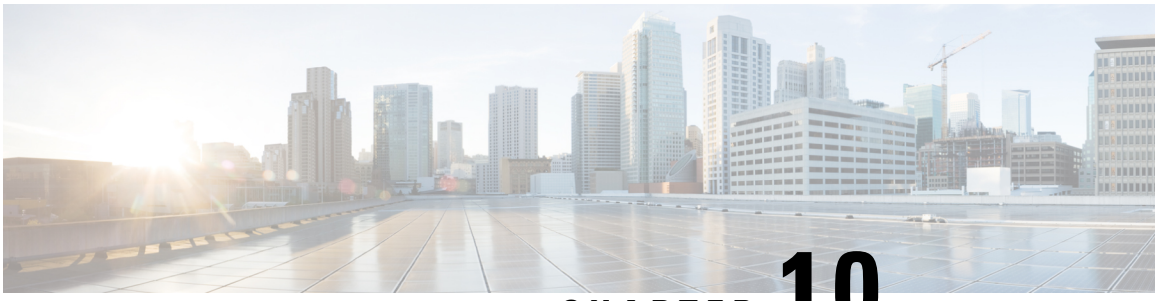
Feature Information for Configuring Authorization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 22: Feature Information for Configuring Authorization

Feature Name	Releases	Feature Information
Named Method Lists for AAA Authorization and Accounting	Cisco IOS XE Release 2.1	Method lists for authorization define the ways that authorization will be performed and the sequence in which these methods will be performed. A method list is simply a named list describing the authorization methods to be queried (such as RADIUS or TACACS+), in sequence. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 10

Configuring Accounting

The AAA accounting feature allows the services that users are accessing and the amount of network resources that users are consuming to be tracked. When AAA accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method is implemented) in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, and auditing.

Prerequisites for Configuring Accounting

The following tasks must be performed before configuring accounting using named method lists:

- Enable AAA on the network access server.
- Define the characteristics of the RADIUS or TACACS+ security server if RADIUS or TACACS+ authorization is issued. For more information about configuring the Cisco network access server to communicate with the RADIUS security server, see the chapter Configuring RADIUS. For more information about configuring the Cisco network access server to communicate with the TACACS+ security server, see the chapter Configuring TACACS+.

Restrictions for Configuring Accounting

The AAA Accounting feature has the following restrictions:

- Accounting information can be sent simultaneously to a maximum of four AAA servers.
- Service Selection Gateway (SSG) restriction--For SSG systems, the **aaa accounting network broadcast** command broadcasts only **start-stop** accounting records. If interim accounting records are configured

using the**ssg accounting interval** command, the interim accounting records are sent only to the configured default RADIUS server.

Information About Configuring Accounting

Named Method Lists for Accounting

Like authentication and authorization method lists, method lists for accounting define the way accounting is performed and the sequence in which these methods are performed.

Named accounting method lists allow a particular security protocol to be designated and used on specific lines or interfaces for accounting services. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is a named list describing the accounting methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists allow one or more security protocols to be designated and used for accounting, thus ensuring a backup system for accounting if the initial method fails. Cisco IOS XE software uses the first method listed to support accounting; if that method fails to respond, the Cisco IOS XE software selects the next accounting method listed in the method list. This process continues until there is successful communication with a listed accounting method, or all methods defined are exhausted.

Note The Cisco IOS XE software attempts accounting with the next listed accounting method only when there is no response from the previous method. If accounting fails at any point in this cycle--meaning that the security server responds by denying the user access--the accounting process stops and no other accounting methods are attempted.

Accounting method lists are specific to the type of accounting being requested. AAA supports six different types of accounting:

- Network--Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
- EXEC--Provides information about user EXEC terminal sessions of the network access server.
- Command--Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- Connection--Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
- System--Provides information about system-level events.
- Resource--Provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.

Note System accounting does not use named accounting lists; only the default list for system accounting can be defined.

When a named method list is created, a particular list of accounting methods for the indicated accounting type are defined.

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods are performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no accounting takes place.

This section includes the following subsections:

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. The figure below shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

In Cisco IOS XE software, RADIUS and TACACS+ server configurations are global. A subset of the configured server hosts can be specified using server groups. These server groups can be used for a particular service. For example, server groups allow R1 and R2 to be defined as separate server groups (SG1 and SG2), and T1 and T2 as separate server groups (SG3 and SG4). This means either R1 and T1 (SG1 and SG3) can be specified in the method list or R2 and T2 (SG2 and SG4) in the method list, which provides more flexibility in the way that RADIUS and TACACS+ resources are assigned.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service--for example, accounting--the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, see Configuring RADIUS module or Configuring TACACS+ module in the *Cisco IOS XE Security Configuration Guide: Securing User Services* Release 2.

AAA Accounting Methods

Cisco IOS XE supports the following two methods for accounting:

- TACACS+—The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.

- RADIUS—The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.

Note With CSCuc32663, passwords and accounting logs are masked before being sent to the TACACS+ or RADIUS security servers. Use the **aaa accounting commands visible-keys** command to send unmasked information to the TACACS+ or RADIUS security servers.

Accounting Record Types

For minimal accounting, use the **stop-only** keyword, which instructs the specified method (RADIUS or TACACS+) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the **none** keyword.

Accounting Methods

The table below lists the supported accounting keywords.

Table 23: AAA Accounting Methods

Keyword	Description
group radius	Uses the list of all RADIUS servers for accounting.
group tacacs+	Uses the list of all TACACS+ servers for accounting.
group group-name	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group *group-name*.

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

To create a default list that is used when a named list is *not* specified in the **aaa accounting**command, use the **default** keyword followed by the methods that are wanted to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa accounting network default stop-only group radius
```

AAA accounting supports the following methods:

- **group tacacs** --To have the network access server send accounting information to a TACACS+ security server, use the **group tacacs+** *method* keyword.

- **group radius** --To have the network access server send accounting information to a RADIUS security server, use the **group radius** *method* keyword.

Note Accounting method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for accounting applies.

- **group** *group-name* --To specify a subset of RADIUS or TACACS+ servers to use as the accounting method, use the **aaa accounting**command with the **group** *group-name* method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of network accounting when no other method list has been defined, enter the following command:

```
aaa accounting network default start-stop group loginrad
```

Before a group name can be used as the accounting method, communication with the RADIUS or TACACS+ security server must be enabled.

AAA Accounting Types

Named accounting method lists are specific to the indicated type of accounting.

- **network** --To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARAP protocols), use the **network** keyword in the **aaa accounting** command. For example, to create a method list that provides accounting information for ARAP (network) sessions, use the **arap** keyword in the **accounting** command.
- **exec** --To create a method list that provides accounting records about user EXEC terminal sessions on the network access server, including username, date, start and stop times, use the **exec** keyword.
- **commands** --To create a method list that provides accounting information about specific, individual EXEC commands associated with a specific privilege level, use the **commands**keyword.
- **connection** --To create a method list that provides accounting information about all outbound connections made from the network access server, use the **connection** keyword.
- resource --To creates a method list to provide accounting records for calls that have passed user authentication or calls that failed to be authenticated.

Note System accounting does not support named method lists.

Network Accounting

Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through an EXEC session:

```
Wed Jun 27 04:44:45 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:45:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:47:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

```
Wed Jun 27 04:48:45 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "408"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 27 04:00:35 2001 172.16.25.15    username1   tty4    562/4327528     starttask_id=28
      service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15    username1   tty4 562/4327528     starttask_id=30
     addr=10.1.1.1   service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15    username1   tty4    408/4327528     update
task_id=30      addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15    username1   tty4    562/4327528     stoptask_id=30
       addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1   bytes_in=2844
  bytes_out=1682  paks_in=36      paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15    username1   tty4    562/4327528     stoptask_id=28
       service=shell   elapsed_time=57
```

Note The precise format of accounting packets records may vary depending on the security server daemon.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:30:52 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
```

```
        Framed-Protocol = PPP
        Framed-IP-Address = "10.1.1.1"
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:02:19 2001 172.16.25.15    username1   Async5  562/4327528     starttask_id=35
      service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    username1   Async5  562/4327528     update
task_id=35      service=ppp     protocol=ip     addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15    username1   Async5  562/4327528     stoptask_id=35
       service=ppp     protocol=ip     addr=10.1.1.2   bytes_in=3366   bytes_out=2149
  paks_in=42      paks_out=28     elapsed_time=164
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:27:25 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 27 03:46:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=2       service=shell
Wed Jun 27 04:08:55 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=2       service=shell   elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:48:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:06:53 2001        172.16.25.15    username1   tty26   10.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    username1   tty26   10.68.202.158
stoptask_id=41       service=shell   elapsed_time=9
```

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 27 03:46:47 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=4       service=shell   priv-lvl=1      cmd=show interfaces <cr>
Wed Jun 27 03:47:03 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 27 03:47:17 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=7       service=shell   priv-lvl=15     cmd=interface GigabitEthernet0/0/0
 <cr>
Wed Jun 27 03:47:29 2001        172.16.25.15    username1   tty3    56223294304327528  stop
     task_id=8       service=shell   priv-lvl=15     cmd=ip address 10.1.1.1 255.255.255.0
 <cr>
```

Note The Cisco Systems implementation of RADIUS does not support command accounting.

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, LAT, TN3270, PAD, and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 03:47:43 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=10     service=connection     protocol=telnet addr=10.68.202.158 cmd=telnet
 username1-sun
Wed Jun 27 03:48:38 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=10     service=connection     protocol=telnet addr=10.68.202.158 cmd=telnet
 username1-sun     bytes_in=4467   bytes_out=96    paks_in=61     paks_out=72 elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "10.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:30:09 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "10.68.202.158"
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 03:48:46 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=12     service=connection     protocol=rlogin addr=10.68.202.158 cmd=rlogin
 username1-sun /user username1
Wed Jun 27 03:51:37 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=12     service=connection     protocol=rlogin addr=10.68.202.158 cmd=rlogin
 username1-sun /user username1 bytes_in=659926 bytes_out=138   paks_in=2378    paks_
out=1251        elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 27 03:53:06 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX
Wed Jun 27 03:54:15 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX  bytes_in=0      bytes_out=0     paks_in=0      paks_out=0      elapsed_time=6
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA accounting has been turned off:

```
Wed Jun 27 03:55:32 2001        172.16.25.15    unknown unknown unknown start   task_id=25
   service=system  event=sys_acct  reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on the TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA accounting has been turned on:

```
Wed Jun 27 03:55:22 2001        172.16.25.15    unknown unknown unknown stop    task_id=23
   service=system  event=sys_acct  reason=reconfigure
```

Additional tasks for measuring system resources are covered in the Cisco IOS XE software configuration guides. For example, IP accounting tasks are described in the Configuring IP Services chapter in the *CiscoIOS XE Application Services Configuration Guide,* Release 2.

Resource Accounting

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. The additional feature of generating “stop” records for calls that fail to authenticate as part of user authentication is also supported. Such records are necessary for users employing accounting records to manage and monitor their networks.

This section includes the following subsections:

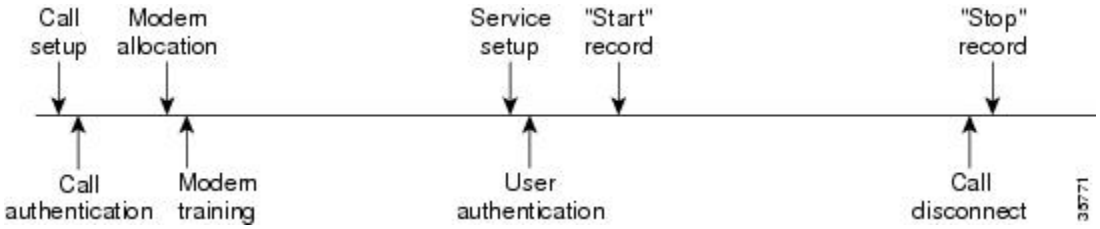
AAA Resource Failure Stop Accounting

Before AAA resource failure stop accounting, there was no method of providing accounting records for calls that failed to reach the user authentication stage of a call setup sequence. Such records are necessary for users employing accounting records to manage and monitor their networks and their wholesale customers.

This functionality generates a “stop” accounting record for any calls that do not reach user authentication; “stop” records are generated from the moment of call setup. All calls that pass user authentication behave as they did before; that is, no additional accounting records are seen.

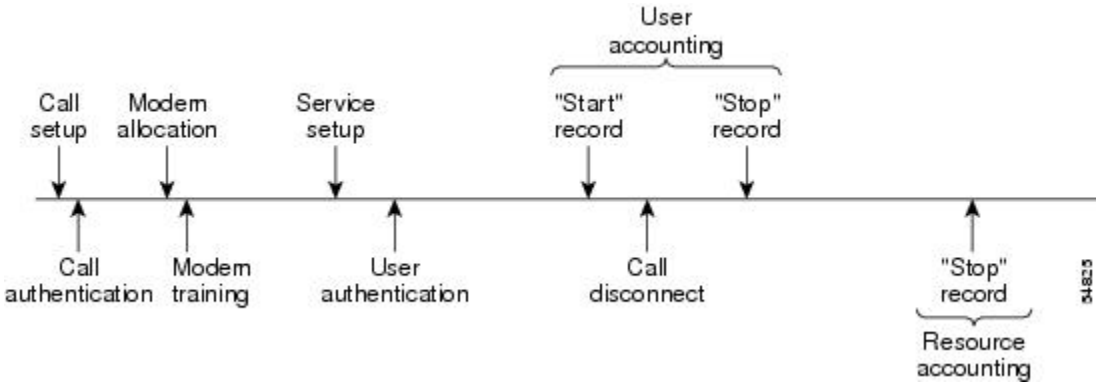
The figure below illustrates a call setup sequence with normal call flow (no disconnect) and without AAA resource failure stop accounting enabled.

Figure 4: Modem Dial-In Call Setup Sequence with Normal Flow and Without Resource Failure Stop Accounting Enabled



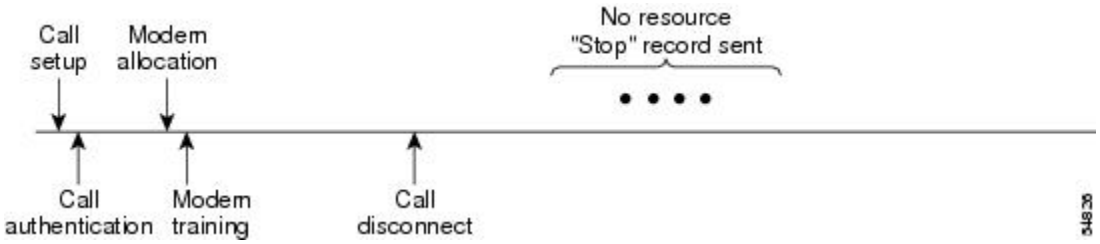
The figure below illustrates a call setup sequence with normal call flow (no disconnect) and with AAA resource failure stop accounting enabled.

Figure 5: Modem Dial-In Call Setup Sequence with Normal Flow and wIth Resource Failure Stop Accounting Enabled



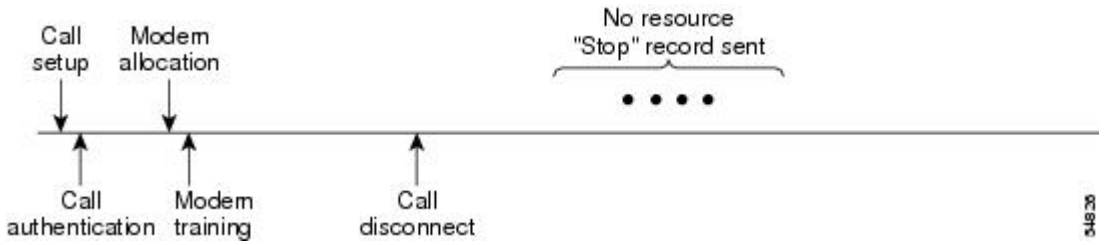
The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and with AAA resource failure stop accounting enabled.

Figure 6: Modem Dial-In Call Setup Sequence with Call Disconnect Occurring Before User Authentication and with Resource Failure Stop Accounting Enabled



The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and without AAA resource failure stop accounting enabled.

Figure 7: Modem Dial-In Call Setup Sequence with Call Disconnect Occurring Before User Authentication and Without Resource Failure Stop Accounting Enabled



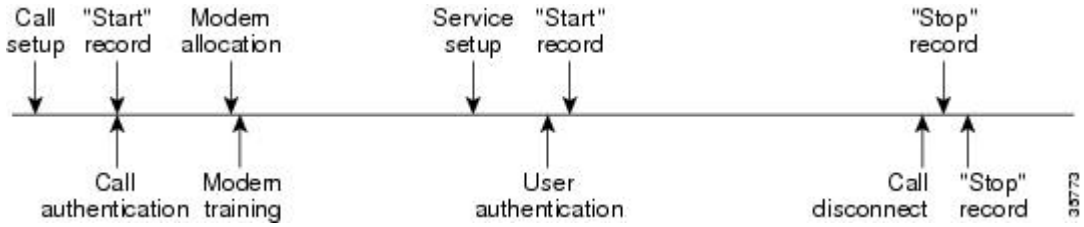
AAA Resource Accounting for Start-Stop Records

AAA resource accounting for start-stop records supports the ability to send a “start” record at each call setup, followed by a corresponding “stop” record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.

With this feature, a call setup and call disconnect “start-stop” accounting record tracks the progress of the resource connection to the device. A separate user authentication “start-stop” accounting record tracks the user management progress. These two sets of accounting records are interlinked by using a unique session ID for the call.

The figure below illustrates a call setup sequence with AAA resource start-stop accounting enabled.

Figure 8: Modem Dial-In Call Setup Sequence with Resource Start-Stop Accounting Enabled



AAA Accounting Enhancements

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

- Statistics for each AAA function (when used in conjunction with the show radius statistics command)
- Status of servers providing AAA functions
- Identities of external AAA servers
- Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether to terminate an active call

The table below shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 24: SNMP End-User Data Objects

Field	Descriptions
SessionId	The session identification used by the AAA accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.
IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

The table below describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 25: SNMP AAA Session Summary

Field	Descriptions
ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present since last system reinstallation.
TotalSessions	Total number of sessions since the last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected since the last system reinstallation.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ attribute-value (AV) pairs or RADIUS attributes, depending on which security method is implemented.

How to Configure AAA Accounting

Configuring AAA Accounting Using Named Method Lists

To configure AAA accounting using named method lists, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2*...]]
2. **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
3. **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2*...]]	Creates an accounting method list and enables accounting. The *list-name* argument is a character string used to name the created list.
Step 2	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] **Example:** `Router(config)# interface interface-type interface-number`	Enters line configuration mode for the lines to which the accounting method list is applied or enters interface configuration mode for the interfaces to which the accounting method list is applied.
Step 3	**accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*} **Example:** `Router(config-if)# ppp accounting {default | list-name}`	Applies the accounting method list to a line or set of lines or applies the accounting method list to an interface or set of interfaces.

What to do next

Note System accounting does not use named method lists. For system accounting, define only the default method list.

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA accounting is activated, the Cisco IOS XE software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the **aaa authentication login** *method-list* **none** command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command or Action	Purpose
`Router(config)# aaa accounting suppress null-username`	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command or Action	Purpose
`Router(config)# aaa accounting update [newinfo] [periodic] number`	Enables periodic interim accounting records to be sent to the accounting server.

When the **aaa accounting update**commandis activated, the Cisco IOS XE software issues interim accounting records for all users on the system. If the **newinfo** keyword is used, interim accounting records are sent to the accounting server every time there is new accounting information to report. An example of this would be when Internet Protocol Control Protocol (IPCP) completes IP address negotiation with the remote peer. The interim accounting record includes the negotiated IP address used by the remote peer.

When **aaa accounting update**command is used with the keyword **periodic**, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

Caution Using the **aaa accounting update periodic** command can cause heavy congestion when many users are logged in to the network.

Configuring an Alternate Method to Enable Periodic Accounting Records

You can use the following alternative method to enable periodic interim accounting records to be sent to the accounting server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network default**
4. **action-type** {**none** | **start-stop** [**periodic** {**disable** | **interval** *minutes*}] | **stop-only**}

5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting network default** **Example:** `Router(config)# aaa accounting network default`	Configures the default accounting for all network-related service requests and enters accounting method list configuration mode.

Step 4	**action-type** {**none** | **start-stop** [**periodic** {**disable** | **interval** *minutes*}] | **stop-only**}
	Example:
	`Router(cfg-acct-mlist)# action-type start-stop`
	Example:
	`periodic interval 5`
		Specifies the type of action to be performed on accounting records.
		• (Optional) The periodic keyword specifies periodic accounting action.
		• The interval keyword specifies the periodic accounting interval.
		• The value argument specifies the intervals for accounting update records (in minutes).
		• The disable keyword disables periodic accounting.

Step 5	**exit** **Example:** `Router(cfg-acct-mlist)# exit`	Returns to global configuration mode.

Generating Interim Service Accounting Records

Perform this task to enable the generation of interim service accounting records at periodic intervals for subscribers.

Before you begin

RADIUS Attribute 85 in the user service profile always takes precedence over the configured interim-interval value. RADIUS Attribute 85 must be in the user service profile. See the RADIUS Attributes Overview and RADIUS IETF Attributes feature document for more information.

Note If RADIUS Attribute 85 is not in the user service profile, then the interim-interval value configured in Generating Interim Accounting Records is used for service interim accounting records.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **subscriber service accounting interim-interval** *minutes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**subscriber service accounting interim-interval** *minutes* **Example:** `Router(config)# subscriber service accounting interim-interval 10`	Enables the generation of interim service accounting records at periodic intervals for subscribers. The *minutes* argument indicates the number of periodic intervals to send accounting update records from 1 to 71582 minutes.

Generating Accounting Records for a Failed Login or Session

When AAA accounting is activated, the Cisco IOS XE software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting send stop-record authentication failure	Generates “stop” records for users who fail to authenticate at login or during session negotiation using PPP.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, it can be specified that NETWORK records be generated before EXEC-stop records. In some cases, such as billing customers for specific services, is can be desirable

to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the network accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting nested	Nests network accounting records.

Suppressing System Accounting Records over Switchover

To suppress the system accounting-on and accounting-off messages during switchover, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting redundancy suppress system-records	Suppresses the system accounting messages during switchover.

Configuring AAA Resource Failure Stop Accounting

To enable resource failure stop accounting, use the following command in global configuration:

Command or Action	Purpose
aaa accounting resource *method-list* **stop-failure group** *server-group*	Generates a "stop" record for any calls that do not reach user authentication. **Note** Before configuring the AAA Resource Failure Stop Accounting feature, the tasks described in the Prerequisites for Configuring Accounting, on page 133 section must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco ASR 1000 Series Aggregation Services Router, see the Configuring SNMP Support chapter in the Cisco IOS XE Network Management Configuration Guide.

Configuring AAA Resource Accounting for Start-Stop Records

To enable full resource accounting for start-stop records, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting resource *method-list* **start-stop group** *server-group*	Supports the ability to send a "start" record at each call setup. followed with a corresponding "stop" record at the call disconnect. **Note** Before configuring this feature, the tasks described in the section Prerequisites for Configuring Accounting, on page 133 must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco ASR 1000 Series Aggregation Services Router, see the chapter Configuring SNMP Support in the Cisco IOS XE Network Management Configuration Guide, Release 2.

Configuring AAA Broadcast Accounting

To configure AAA broadcast accounting, use the **aaa accounting** command in global configuration mode. This command has been modified to allow the **broadcast** keyword.

Command or Action	Purpose
aaa accounting {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] *method1* [*method2...*]	Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per DNIS, use the **aaa dnis map accounting network**command in global configuration mode. This command has been modified to allow the **broadcast** keyword and multiple server groups.

Command or Action	Purpose
aaa dnis map *dnis-number* **accounting network** [**start-stop** | **stop-only** | **none**] [**broadcast**] *method1* [*method2...*]	Allows per-DNIS accounting configuration. This command has precedence over the global **aaa accounting** command. Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring the AAA Session MIB

The following tasks must be performed before configuring the AAA session MIB feature:

- Configure SNMP. For information on SNMP, see the Configuring SNMP Support chapter in the Cisco IOS XE Network Management Configuration Guide.
- Configure AAA.

- Define the RADIUS or TACACS+ server characteristics.

Note Overusing SNMP can affect the overall system performance; therefore, normal network management performance must be considered when this feature is used.

To configure the AAA session MIB, use the following command in global configuration mode:

Command or Action	Purpose
aaa session-mib disconnect	Monitors and terminates authenticated client connections using SNMP. To terminate the call,use the disconnect keyword .

Establishing a Session with a Router if the AAA Server Is Unreachable

To establish a console session with a router if the AAA server is unreachable, use the following command in global configuration mode:

Command or Action	Purpose
no aaa accounting system guarantee-first	The **aaa accounting system guarantee-first** command guarantees system accounting as the first record, which is the default condition. In some situations, users may be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than three minutes. To resolve this problem, use the **no aaa accounting system guarantee-first** command.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users logged in, use the following command in privileged EXEC mode:

Command or Action	Purpose
show accounting	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command or Action	Purpose
debug aaa accounting	Displays information on accountable events as they occur.

Configuration Examples for AAA Accounting

Configuring a Named Method List Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) in order for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database is queried for authentication and authorization information, and accounting services are handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network network1 group radius local
aaa accounting network network2 start-stop group radius group tacacs+
username root password ALongPassword
tacacs-server host 172.31.255.0
tacacs-server key goaway
radius-server host 172.16.2.7
radius-server key myRaDiUSpassWoRd
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization network1
 ppp accounting network2
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins”, which specifies that first RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.
- The **aaa authorization network network1 group radius local** command defines the network authorization method list named “network1”, which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.
- The **aaa accounting network network2 start-stop group radius group tacacs+** command defines the network accounting method list named “network2”, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP. If the RADIUS server fails to respond, accounting services are handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs-server host** command defines the name of the TACACS+ server host.

- The **tacacs-server key** command defines the shared secret text string between the network access server and the TACACS+ server host.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The**encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The**ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The**ppp authorization network1** command applies the blue1 network authorization method list to the specified interfaces.
- The**ppp accounting network2** command applies the red1 network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS XE software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to accept only incoming calls.

The **show accounting** command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User username2 Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

The table below describes the fields contained in the preceding output.

Table 26: show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User’s ID.
Priv	User’s privilege level.
Task ID	Unique identifier for each accounting session.

Field	Description
Accounting Record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.

Configuring AAA Resource Accounting Example

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method
 to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all start-stop
 accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to
 use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

Configuring AAA Broadcast Accounting Example

The following example shows how to turn on broadcast accounting using the global **aaa accounting** command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2
aaa group server tacacs+ isp_customer
 server 172.0.0.1
aaa accounting network default start-stop broadcast group isp group isp_customer
radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key1
tacacs-server host 172.0.0.1 key key2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connections to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Configuring per-DNIS AAA Broadcast Accounting Example

The following example shows how to turn on per-DNIS broadcast accounting using the global **aaa dnis map accounting network**command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2
aaa group server tacacs+ isp_customer
 server 172.0.0.1
aaa dnis map enable
aaa dnis map 7777 accounting network start-stop broadcast group isp group isp_customer
radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key_1
tacacs-server host 172.0.0.1 key key_2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connection calls having DNIS number 7777 to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

AAA Session MIB Example

The following example shows how to set up the AAA session MIB feature to disconnect authenticated client connections for PPP users:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
aaa session-mib disconnect
```

Additional References

The following sections provide references related to the Configuring Accounting feature.

Related Documents

Related Topic	Document Title
Configuring SNMP	*Cisco IOS XE Network Management Configuration Guide*
SNMP commands	*Cisco IOS Network Management Command Reference*
Security commands	*Cisco IOS Security Command Reference*
Configuring Radius	Configuring RADIUS
Configuring TACACS+	Configuring TACACS+
Configuring IP Services	*Cisco IOS XE Application Services Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-AAA-SESSION-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases , and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

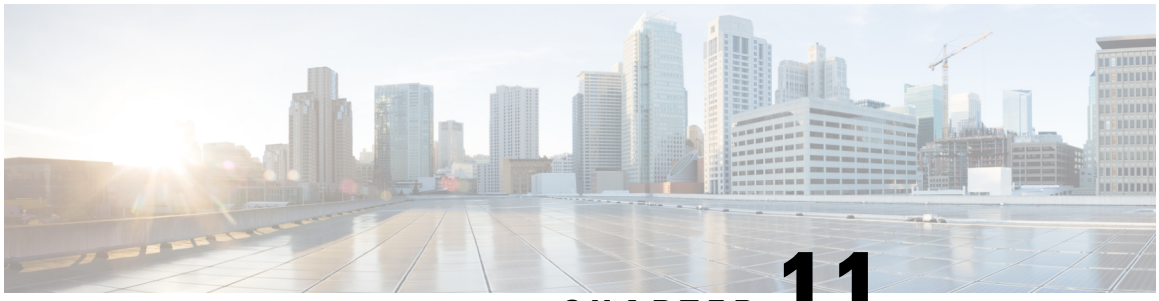
Feature Information for Configuring Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 27: Feature Information for Configuring Accounting

Feature Name	Releases	Feature Information
AAA Broadcast Accounting	Cisco IOS XE Release 2.1	AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa accounting**.
AAA Session MIB	Cisco IOS XE Release 2.1	The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa session-mib disconnect**.
Connection Accounting	Cisco IOS XE Release 2.1	Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
AAA Interim Accounting	Cisco IOS XE Release 2.4	AAA interim accounting allows accounting records to be sent to the accounting server every time there is new accounting information to report, or on a periodic basis. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa accounting update** and **subscriber service accounting interim-interval**.



CHAPTER 11

AAA-SERVER-MIB Set Operation

The AAA-SERVER-MIB Set Operation feature allows you to extend and expand your ability to configure authentication, authorization, and accounting (AAA) servers using the CISCO-AAA-SERVER-MIB. Using this feature, you can do the following:

- Create and add new AAA servers.
- Modify the "KEY" under the CISCO-AAA-SERVER-MIB.
- Delete the AAA server configuration.

Prerequisites for AAA-SERVER-MIB Set Operation

AAA must have been enabled on the router, that is, the **aaa new-model** command must have been configured. If this configuration has not been accomplished, the set operation fails.

Restrictions for AAA-SERVER-MIB Set Operation

Currently, the CISCO SNMP set operation is supported only for the RADIUS protocol. Therefore, only RADIUS servers in global configuration mode can be added, modified, or deleted.

Information About AAA-SERVER-MIB Set Operation

CISCO-AAA-SERVER-MIB

The CISCO-AAA-SERVER-MIB provides that statistics reflect both the state of the AAA server operation with the server itself and of AAA communications with external servers. The CISCO-AAA-SERVER-MIB provides the following information:

- Statistics for each AAA operation
- Status of servers that are providing AAA functions
- Identities of external AAA servers

CISCO-AAA-SERVER-MIB Set Operation

In Cisco IOS XE Release 2.1, the CISCO-AAA-SERVER-MIB supports both the get and set operations. With the set operation, you can do the following:

- Create or add a new AAA server.
- Modify the KEY under the CISCO-AAA-SERVER-MIB. This "secret key" is used for secure connectivity to the AAA server, which is present with the network access server (NAS) and the AAA server.
- Delete the AAA server configuration.

How to Configure AAA-SERVER-MIB Set Operation

No special configuration is required for this feature. The Simple Network Management Protocol (SNMP) framework can be used to manage MIBs. See the section Additional References for a reference to configuring SNMP.

Verifying RADIUS Server Configuration and Server Statistics

RADIUS server configuration and server statistics can be verified by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config | include radius-server host**
3. **show aaa servers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**show running-config | include radius-server host** **Example:** `Router# show running-config | include radius-server host`	Displays all the RADIUS servers that are configured in the global configuration mode.
Step 3	**show aaa servers** **Example:** `Router# show aaa servers`	Displays information about the number of requests sent to and received from authentication, authorization, and accounting (AAA) servers.

Configuration Examples for AAA-SERVER-MIB Set Operation

RADIUS Server Configuration and Server Statistics Example

The following output example shows the RADIUS server configuration and server statistics before and after the set operation.

Before the Set Operation

```
Router# show running-config | include radius-server host
! The following line is for server 1.
radius-server host 172.19.192.238 auth-port 2095 acct-port 2096 key cisco2
! The following line is for server 2.
radius-server host 172.19.192.238 auth-port 1645 acct-port 1646
```

Server Statistics

```
Router# show aaa servers
RADIUS: id 2, priority 1, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 25s, previous duration 0s
     Dead: total time 0s, count 7
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 2
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 5m
RADIUS: id 3, priority 2, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 5s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
```

```
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 3m
```

SNMP Get Operation to Check the Configuration and Statistics of the RADIUS Servers

```
aaa-server5:/users/smetri> getmany 10.0.1.42 casConfigTable
casAddress.2.2 = 172.19.192.238
casAddress.2.3 = 172.19.192.238
casAuthenPort.2.2 = 2095
casAuthenPort.2.3 = 1645
casAcctPort.2.2 = 2096
casAcctPort.2.3 = 1646
casKey.2.2 =
casKey.2.3 =
! The following line shows priority for server 1.
casPriority.2.2 = 1
! The following line shows priority for server 2.
casPriority.2.3 = 2
casConfigRowStatus.2.2 = active(1)
casConfigRowStatus.2.3 = active(1)
aaa-server5:/users/smetri>
```

SNMP Set Operation

The key of the existing RADIUS server is being changed. The index “1” is being used. That index acts as a wildcard for addition, deletion, or modification of any entries.

```
Change the key for server 1:=>
aaa-server5:/users/smetri>  setany -v2c 10.0.1.42 public casAddress.2.1 -a 172.19.192.238
casAuthenPort.2.1 -i 2095 casAcctPort.2.1 -i 2096 casKey.2.1 -o king
casAddress.2.1 = 172.19.192.238
casAuthenPort.2.1 = 2095
casAcctPort.2.1 = 2096
casKey.2.1 = king
aaa-server5:/users/smetri>
```

After the Set Operation

After the above SNMP set operation, the configurations on the router change. The following output shows the output after the set operation.

```
Router# show running-config | include radius-server host
radius-server host 172.19.192.238 auth-port 1645 acct-port 1646
! The following line shows a change in the key value to “king.”
radius-server host 172.19.192.238 auth-port 2095 acct-port 2096 key king
```

```
Router# show aaa servers
RADIUS: id 3, priority 1, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 189s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
```

```
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 6m

! The following line shows a new server with new statistics.
RADIUS: id 4, priority 2, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 209s, previous duration 0s
     Dead: total time 0s, count 7
Authen: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Authentication commands	*Cisco IOS Security Command Reference*
IEEE 802.1x—Flexible Authentication	*Securing User Services Configuration Library*

MIBs

MIB	MIBs Link
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

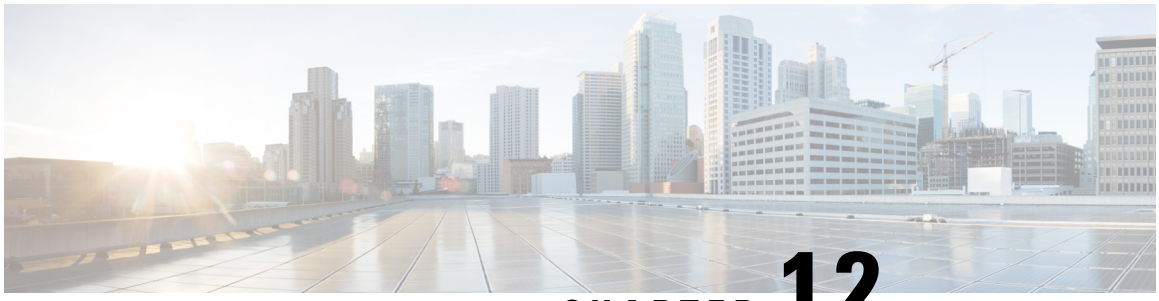
Feature Information for AAA-SERVER-MIB Set Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 28: Feature Information for AAA-SERVER-MIB Set Operation

Feature Name	Releases	Feature Information
AAA-SERVER-MIB Set Operation	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 12

Per VRF AAA

The Per VRF AAA feature allows ISPs to partition authentication, authorization, and accounting (AAA) services on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances, allowing their customers to control some of their own AAA services.

The list of servers in server groups is extended to include the definitions of private servers in addition to references to the hosts in the global configuration, allowing access to both customer servers and global service provider servers simultaneously.

In Cisco IOS XE Release 2.4 and later releases, a customer template can be used, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. This feature is referred to as the Dynamic Per VRF AAA feature.

Prerequisites for Per VRF AAA

Before configuring the Per VRF AAA feature, AAA must be enabled. See “How to Configure Per VRF AAA” section on page 6 for more information.

Restrictions for Per VRF AAA

- This feature is supported only for RADIUS servers.
- Operational parameters should be defined once per VRF rather than set per server group, because all functionality must be consistent between the network access server (NAS) and the AAA servers.
- The ability to configure a customer template either locally or remotely is available only for Cisco IOS XE Release 2.4 and later releases.

Information About Per VRF AAA

When you use the Per VRF AAA feature, AAA services can be based on VRF instances. This feature permits the Provider Edge (PE) or Virtual Home Gateway (VHG) to communicate directly with the customer's RADIUS server, which is associated with the customer's Virtual Private Network (VPN), without having to go through a RADIUS proxy. Thus, ISPs can scale their VPN offerings more efficiently because they no longer have to use RADIUS proxies and ISPs can also provide their customers with additional flexibility.

How Per VRF AAA Works

To support AAA on a per customer basis, some AAA features must be made VRF aware. That is, ISPs must be able to define operational parameters--such as AAA server groups, method lists, system accounting, and protocol-specific parameters--and bind those parameters to a particular VRF instance. Defining and binding the operational parameters can be accomplished using one or more of the following methods:

- Virtual private dialup network (VPDN) virtual template or dialer interfaces that are configured for a specific customer
- Locally defined customer templates--Per VPN with customer definitions. The customer template is stored locally on the VHG. This method can be used to associate a remote user with a specific VPN based on the domain name or dialed number identification service (DNIS) and provide the VPN-specific configuration for virtual access interface and all operational parameters for the customer AAA server.
- Remotely defined customer templates--Per VPN with customer definitions that are stored on the service provider AAA server in a RADIUS profile. This method is used to associate a remote user with a specific VPN based on the domain name or DNIS and provide the VPN-specific configuration for the virtual access interface and all operational parameters for the AAA server of the customer.

Note The ability to configure locally or remotely defined customer templates is available only with Cisco IOS XE Release 2.4 and later releases.

AAA Accounting Records

The Cisco implementation of AAA accounting provides "start" and "stop" record support for calls that have passed user authentication. Start and stop records are necessary for users employing accounting records to manage and monitor their networks.

New Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (VSA) attribute 26. Attribute 26 encapsulates VSAs, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This format allows the full set of features available for TACACS+ authorization to be used also for RADIUS.

The table below summarizes the VSAs that are now supported with Per VRF AAA.

Table 29: VSAs Supported with Per VRF AAA

VSA Name	Value Type	Description
Note Each VSA must have the prefix "template:" before the VSA name, unless a different prefix is explicitly stated.		
account-delay	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting delay-start**command for the customer template.
account-send-stop	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting send stop-record authentication**command with the **failure** keyword.
account-send-success-remote	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting send stop-record authentication**command with the **success** keyword.
attr-44	string	This VSA must be "access-req." The functionality of this VSA is equal to the **radius-server attribute 44 include-in-access-req** command.
ip-addr	string	This VSA specifies the IP address, followed by the mask that the router uses to indicate its own IP address and mask in negotiation with the client; for example, ip-addr=192.168.202.169 255.255.255.255
ip-unnumbered	string	This VSA specifies the name of an interface on the router. The functionality of this VSA is equal to the **ip unnumbered** command, which specifies an interface name such as "Loopback 0."
ip-vrf	string	This VSA specifies which VRF will be used for the packets of the end user. This VRF name should match the name that is used on the router via the **ip vrf forwarding** command.

VSA Name	Value Type	Description
peer-ip-pool	string	This VSA specifies the name of an IP address pool from which an address will be allocated for the peer. This pool should be configured using the **ip local pool**command or should be automatically downloadable via RADIUS.
ppp-acct-list	string	This VSA defines the accounting method list that is to be used for PPP sessions. The VSA syntax is as follows: “ppp-acct-list=[start-stop | stop-only | none] group X [group Y] [broadcast].” It is equal to the **aaa accounting network mylist** command functionality. The user must specify at least one of the following options: start-stop, stop-only, or none. If either start-stop or stop-only is specified, the user must specify at least one, but not more than four, group arguments. Each group name must consist of integers. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.” After each group has been specified, the user can specify the broadcast option
ppp-authen-list	string	This VSA defines which authentication method list is to be used for PPP sessions and, if more than one method is specified, in what order the methods should be used. The VSA syntax is as follows: “ppp-authen-list=[groupX | local | local-case | none | if-needed],” which is equal to the **aaa authentication ppp mylist** command functionality. The user must specify at least one, but no more than four, authentication methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.”
ppp-authen-type	string	This VSA allows the end user to specify at least one of the following authentication types: pap, chap, eap, ms-chap, ms-chap-v2, any, or a combination of the available types that is separated by spaces. The end user will be permitted to log in using only the methods that are specified in this VSA. PPP will attempt these authentication methods in the order presented in the attribute.

VSA Name	Value Type	Description
ppp-author-list	string	This VSA defines the authorization method list that is to be used for PPP sessions. It indicates which methods will be used and in what order. The VSA syntax is as follows: "ppp-author-list=[groupX] [local] [if-authenticated] [none]," which is equal to the **aaa authorization network mylist**command functionality. The user must specify at least one, but no more than four, authorization methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA "rad-serv."
Note The RADIUS VSAs--rad-serv, rad-serv-filter, rad-serv-source-if, and rad-serv-vrf--must have the prefix "aaa:" before the VSA name.		
rad-serv	string	This VSA indicates the IP address, key, timeout, and retransmit number of a server, as well as the group of the server. The VSA syntax is as follows: "rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W]." Other than the IP address, all parameters are optional and can be issued in any order. If the optional parameters are not specified, their default values will be used. The key cannot contain any spaces; for "retransmit V," "V" can range from 1-100; for "timeout W," the "W" can range from 1-1000.
rad-serv-filter	string	The VSA syntax is as follows: "rad-serv-filter=authorization | accounting-request | reply-accept | reject-filtername." The filtername must be defined via the **radius-server attribute list filtername** command. **Note** This VSA is supported in Cisco IOS XE Release 2.3 and later releases.
rad-serv-source-if	string	This VSA specifies the name of the interface that is used for transmitting RADIUS packets. The specified interface must match the interface configured on the router.

VSA Name	Value Type	Description
rad-serv-vrf	string	This VSA specifies the name of the VRF that is used for transmitting RADIUS packets. The VRF name should match the name that was specified via the **ip vrf forwarding** command.

VRF Aware Framed-Routes

In Cisco IOS XE Release 2.3 and later, the Cisco ASR 1000 Series Aggregation Services Routers support VRF aware framed-routes. No configuration is required to enable support for this feature. Framed-routes are automatically detected and if the framed-route is part of a VRF associated with an interface, the rout is applied accordingly.

How to Configure Per VRF AAA

Configuring Per VRF AAA

Configuring AAA

To enable AAA you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.

Configuring Server Groups

To configure server groups you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *groupname*
5. **server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa group server radius** *groupname* **Example:** `Router(config)# aaa group server radius v2.44.com`	Groups different RADIUS server hosts into distinct lists and distinct methods. Enters server-group configuration mode.
Step 5	**server-private** *ip-address* [**auth-port** *port-number*	**acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] **Example:** `Router(config-sg-radius)# server-private 10.10.130.2 auth-port 1600 key ww`	Configures the IP address of the private RADIUS server for the group server. **Note** If private server parameters are not specified, global configurations will be used. If global configurations are not specified, default values will be used.
Step 6	**exit** **Example:**	Exits from server-group configuration mode; returns to global configuration mode.

	Command or Action	Purpose
	`Router(config-sg-radius)# exit`	

Configuring Authentication Authorization and Accounting for Per VRF AAA

To configure authentication, authorization, and accounting for Per VRF AAA, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]
6. **aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
7. **aaa accounting delay-start** [**vrf** *vrf-name*]
8. **aaa accounting send stop-record authentication** {**failure** | **success remote-server**} [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp method_list_v2.44.com group v2.44.com`	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.

	Command or Action	Purpose
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authorization network method_list_v2.44.com group v2.44.com`	Sets parameters that restrict user access to a network.
Step 6	**aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config)# aaa accounting system default vrf v2.44.com start-stop group v2.44.com`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS.
Step 7	**aaa accounting delay-start** [**vrf** *vrf-name*] **Example:** `Router(config)# aaa acounting delay-start vrf v2.44.com`	Displays generation of the start accounting records until the user IP address is established.
Step 8	**aaa accounting send stop-record authentication** {**failure** | **success remote-server**} [**vrf** *vrf-name*] **Example:** `Router(config)# aaa accounting send stop-record authentication failure vrf v2.44.com`	Generates accounting stop records. When using the **failure** keyword a "stop" record will be sent for calls that are rejected during authentication. When using the **success** keyword a "stop" record will be sent for calls that meet one of the following criteria: • Calls that are authenticated by a remote AAA server when the call is terminated. • Calls that are not authenticated by a remote AAA server and the start record has been sent. • Calls that are successfully established and then terminated with the "stop-only" **aaa accounting** configuration. **Note** The **success** and **remote-server** keywords are available in Cisco IOS XE Release 2.4 and later releases.

Configuring RADIUS-Specific Commands for Per VRF AAA

To configure RADIUS-specific commands for Per VRF AAA you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*]
4. **radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*] **Example:** `Router(config)# ip radius source-interface loopback55`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets and enables the specification on a per-VRF basis.
Step 4	**radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*] **Example:** `Router(config)# radius-server attribute 44 include-in-access-req vrf v2.44.com`	Sends RADIUS attribute 44 in access request packets before user authentication and enables the specification on a per-VRF basis.

Configuring Interface-Specific Commands for Per VRF AAA

To configure interface-specific commands for Per VRF AAA, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip vrf forwarding** *vrf-name*
5. **ppp authentication** {*protocol1* [*protocol2...*]} *listname*
6. **ppp authorization** *list-name*
7. **ppp accounting default**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface loopback11`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding v2.44.com`	Associates a VRF with an interface.
Step 5	**ppp authentication** {*protocol1* [*protocol2...*]} *listname* **Example:** `Router(config-if)# ppp authentication chap callin V2_44_com`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 6	**ppp authorization** *list-name* **Example:** `Router(config-if)# ppp authorization V2_44_com`	Enables AAA authorization on the selected interface.
Step 7	**ppp accounting default** **Example:** `Router(config-if)# ppp accounting default`	Enables AAA accounting services on the selected interface.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode.

Configuring Per VRF AAA Using Local Customer Templates

Configuring AAA

Perform the tasks as outlined in the Configuring Per VRF AAA.

Configuring Server Groups

Perform the tasks as outlined in the Configuring Server Groups.

Configuring Authentication Authorization and Accounting for Per VRF AAA

Perform the tasks as outlined in the Configuring Authentication Authorization and Accounting for Per VRF AAA.

Configuring Authorization for Per VRF AAA with Local Customer Templates

To configure authorization for Per VRF AAA with local templates, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables the use of local or remote templates.
Step 4	**aaa authorization network default local** **Example:** `Router(config)# aaa authorization network default local`	Specifies local as the default method for authorization.

Configuring Local Customer Templates

To configure local customer templates, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn search-order domain**
4. **template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**]
5. **peer default ip address pool** *pool-name*
6. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
7. **ppp authorization** [**default** | *list-name*]
8. **aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn search-order domain** **Example:** `Router (config)# vpdn search-order domain`	Looks up the profiles based on domain.
Step 4	**template** *name* [**default**	**exit**	**multilink**	**no**	**peer**	**ppp**] **Example:** `Router (config)# template v2.44.com`	Creates a customer profile template and assigns a unique name that relates to the customer that will be receiving it. Enters template configuration mode. **Note** Steps 5, 6, and 7 are optional. Enter **multilink**, **peer**, and **ppp** keywords appropriate to customer application requirements.
Step 5	**peer default ip address pool** *pool-name* **Example:** `Router(config-template)# peer default ip address pool v2_44_com_pool`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.

	Command or Action	Purpose
Step 6	**ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 7	**ppp authorization** [**default** | *list-name*] **Example:** `Router(config-template)# ppp authorization v2_44_com`	(Optional) Sets the PPP link authorization method.
Step 8	**aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config-template)# aaa accounting v2_44_com`	(Optional) Enables AAA operational parameters for the specified customer profile.
Step 9	**exit** **Example:** `Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.

Configuring Per VRF AAA Using Remote Customer Templates

Configuring AAA

Perform the tasks as outlined in the Configuring Per VRF AAA.

Configuring Server Groups

Perform the tasks as outlined in the Configuring Server Groups.

Configuring Authentication for Per VRF AAA with Remote Customer Profiles

To configure authentication for Per VRF AAA with remote customer profiles, you need to perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2...*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# ppp authentication ppp default group radius`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on serial interfaces that are running PPP.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2...*] **Example:** `Router(config)# aaa authorization network default group sp`	Sets parameters that restrict user access to a network.

Configuring Authorization for Per VRF AAA with Remote Customer Profiles

To configuring authorization for Per VRF AAA with remote customer profiles, you need to perform the following step.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization** {**network** \| **exec** \| **commands** *level* \| **reverse-access** \| **configuration**} {**default** \| *list-name*} [[*method1* [*method2...*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables use of local or remote templates.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2...*] **Example:** `Router(config)# aaa authorization network default sp`	Specifies the server group that is named as the default method for authorization.

Configuring the RADIUS Profile on the SP RADIUS Server

Configure the RADIUS profile on the Service Provider (SP) RADIUS server. See the Per VRF AAA Using a Remote RADIUS Customer Template Example for an example of how to update the RADIUS profile.

Verifying VRF Routing Configurations

To verify VRF routing configurations, you need to complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. show ip route vrf *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	show ip route vrf *vrf-name* **Example:** `Router(config)# show ip route vrf northvrf`	Displays the IP routing table associated with a VRF.

Troubleshooting Per VRF AAA Configurations

To troubleshoot the Per VRF AAA feature, use at least one of the following commands in EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authentication`	Displays information on AAA authentication.
`Router# debug aaa authorization`	Displays information on AAA authorization.
`Router# debug ppp negotiation`	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router# debug radius`	Displays information associated with RADIUS.
`Router# debug vpdn event`	Displays Layer 2 Transport Protocol (L2TP) errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router# debug vpdn error`	Displays debug traces for VPN.

Configuration Examples for Per VRF AAA

Per VRF Configuration Examples

Per VRF AAA Example

The following example shows how to configure the Per VRF AAA feature using a AAA server group with associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa accounting delay-start vrf v1.55.com
```

```
aaa accounting send stop-record authentication failure vrf v1.55.com
aaa group server radius v1.55.com
    server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
    ip vrf forwarding v1.55.com
ip radius source-interface loopback55
radius-server attribute 44 include-in-access-req vrf v1.55.com
```

Per VRF AAA Using a Locally Defined Customer Template Example

The following example shows how to configure the Per VRF AAA feature using a locally defined customer template with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa authorization network default local
aaa authorization template
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa group server radius V1_55_com
  server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
  ip vrf forwarding V1.55.com
template V1.55.com
  peer default ip address pool V1_55_com_pool
  ppp authentication chap callin V1_55_com
  ppp authorization V1_55_com
  ppp accounting V1_55_com
  aaa accounting delay-start
  aaa accounting send stop-record authentication failure
  radius-server attribute 44 include-in-access-req
  ip vrf forwarding v1.55.com
  ip radius source-interface Loopback55
```

Per VRF AAA Using a Remote RADIUS Customer Template Example

The following examples shows how to configure the Per VRF AAA feature using a remotely defined customer template on the SP RADIUS server with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization template
aaa authorization network default group sp
aaa group server radius sp
   server 10.3.3.3
radius-server host 10.3.3.3 auth-port 1645 acct-port 1646 key sp_key
```

The following RADIUS server profile is configured on the SP RADIUS server:

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
```

```
framed-protocol = ppp
service-type = framed
```

Customer Template Examples

Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a locally configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authentication ppp V1_55_com group V1_55_com
aaa authorization template
aaa authorization network default local group radius
aaa authorization network V1_55_com group V1_55_com
aaa accounting network V1_55_com start-stop broadcast group V1_55_com group SP_AAA_server
aaa group server radius SP_AAA_server
 server 10.10.100.7 auth-port 1645 acct-port 1646
aaa group server radius V1_55_com
 server-private 10.10.132.4 auth-port 1645 acct-port 1646
 authorization accept min-author
 accounting accept usage-only
 ip vrf forwarding V1.55.com
ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55
template V1.55.com
 peer default ip address pool V1.55-pool
 ppp authentication chap callin V1_55_com
 ppp authorization V1_55_com
 ppp accounting V1_55_com
 aaa accounting delay-start
 aaa accounting send stop-record authentication failure
 radius-server attribute 44 include-in-access-req
vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41
interface Virtual-Template13
 ip vrf forwarding V1.55.com
 ip unnumbered Loopback55
 ppp authentication chap callin
 ppp multilink
ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group
ip radius source-interface Loopback0
ip radius source-interface Loopback55 vrf V1.55.com
radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
radius-server host 10.10.100.7 auth-port 1645 acct-port 1646 key ww
radius-server host 10.10.132.4 auth-port 1645 acct-port 1646 key ww
```

Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a remotely configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authorization template
aaa authorization network default local group radius
ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55
vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41
interface Virtual-Template13
 no ip address
 ppp authentication chap callin
 ppp multilink
ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group
radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
```

The customer template is stored as a RADIUS server profile for v1.55.com.

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "aaa:rad-serv#2=10.10.100.7 key ww"
cisco-avpair = "aaa:rad-serv-source-if#2=Loopback 0"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1 group 2 broadcast"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "aaa:rad-serv-filter#1=authorization accept min-author"
cisco-avpair = "aaa:rad-serv-filter#1=accounting accept usage-only"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

AAA Accounting Stop Record Examples

The following AAA accounting stop record examples show how to configure the **aaa accounting send stop-record authentication** command to control the generation of “stop” records when the **aaa accounting** command is issued with the **start-stop** or **stop-only** keyword.

Note The **success** and **remote-server** keywords are available in Cisco IOS XE Release 2.4 and later releases.

AAA Accounting Stop Record and Rejected Call Example

The following example shows the "stop" record being sent for a rejected call during authentication when the **aaa accounting send stop-record authentication**command is issued with the **success** keyword.

```
Router# show running-config | include aaa
.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication success remote-server
aaa accounting network default start-stop group radius
Router#
*Jul  7 03:39:40.199: AAA/BIND(00000026): Bind i/f Virtual-Template2
*Jul  7 03:39:40.199: ppp21 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:42.199: RADIUS:  AAA Unsupported     [156] 7
*Jul  7 03:39:42.199: RADIUS:   30 2F 30 2F
30                                  [0/0/0]
*Jul  7 03:39:42.199: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026): acct_session_id: 55
*Jul  7 03:39:42.199: RADIUS(00000026): sending
*Jul  7 03:39:42.199: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:42.199: RADIUS(00000026): Send Access-Request to
172.19.192.238:2195 id 1645/14, len 94
*Jul  7 03:39:42.199: RADIUS:  authenticator A6 D1 6B A4 76 9D 52 CF - 33 5D
16 BE AC 7E 5F A6
*Jul  7 03:39:42.199: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.199: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:42.199: RADIUS:  CHAP-Password       [3]   19  *
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:39:42.199: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:42.199: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.199: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:42.271: RADIUS: Received from id 1645/14 172.19.192.238:2195,
Access-Accept, len 194
*Jul  7 03:39:42.271: RADIUS:  authenticator 30 AD FF 8E 59 0C E4 6C - BA 11
23 63 81 DE 6F D7
*Jul  7 03:39:42.271: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  26
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   20  "vpdn:tunnel-
id=lac"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  29
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   23  "vpdn:tunnel-
type=l2tp"
```

```
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  30
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   24  "vpdn:gw-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  31
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   25  "vpdn:nas-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  34
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   28  "vpdn:ip-
addresses=10.0.0.2"
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.275: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.275: RADIUS(00000026): Received from id 1645/14
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-id
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: gw-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: nas-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: ip-addresses
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.279: AAA/BIND(00000027): Bind i/f
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ, flg TLS, ver 2, len 134, tnl
0, ns 0, nr 0
         C8 02 00 86 00 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 09 00 00 00 07 6C 61 63 00 19 00
         00 00 08 43 69 73 63 6F 20 53 79 73 74 65 6D 73
         2C 20 49 6E 63 2E 80 ...
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN, flg TLS, ver 2, len 66, tnl
0, ns 1, nr 0
         C8 02 00 42 00 00 00 00 00 01 00 00 80 08 00 00
         00 00 00 04 80 1E 00 00 00 01 00 02 00 06 54 6F
         6F 20 6D 61 6E 79 20 72 65 74 72 61 6E 73 6D 69
         74 73 00 08 00 09 00 69 00 01 80 08 00 00 00 09
         53 9F
*Jul  7 03:39:49.279: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:49.279: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:49.279: RADIUS(00000026): sending
*Jul  7 03:39:49.279: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:49.279: RADIUS(00000026): Send Accounting-Request to
172.19.192.238:2196 id 1646/32, len 179
*Jul  7 03:39:49.279: RADIUS:  authenticator 0A 85 2F F0 65 6F 25 E1 - 97 54
CC BF EA F7 62 89
*Jul  7 03:39:49.279: RADIUS:  Acct-Session-Id     [44]  10  "00000037"
*Jul  7 03:39:49.279: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                   [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                   [3]
*Jul  7 03:39:49.283: RADIUS:  Acct-Tunnel-Connecti[68]  3   "0"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Client-Auth-I[90]  5   "lac"
*Jul  7 03:39:49.283: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:49.283: RADIUS:  Acct-Authentic      [45]  6
RADIUS                    [1]
*Jul  7 03:39:49.283: RADIUS:  Acct-Session-Time   [46]  6
```

```
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Octets   [42]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Octets  [43]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Packets  [47]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Packets [48]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Terminate-Cause[49]  6   nas-
error                 [9]
*Jul  7 03:39:49.283: RADIUS:  Acct-Status-Type    [40]  6
Stop                      [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:49.283: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:49.283: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:39:49.335: RADIUS: Received from id 1646/32 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:39:49.335: RADIUS:  authenticator C8 C4 61 AF 4D 9F 78 07 - 94 2B
44 44 17 56 EC 03
```

AAA Accounting Stop Record and Successful Call Example

The following example shows “start” and “stop” records being sent for a successful call when the **aaa accounting send stop-record authentication** command is issued with the **failure** keyword.

```
Router# show running-config | include aaa

.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication failure
aaa accounting network default start-stop group radius
.
.
.
*Jul  7 03:28:31.543: AAA/BIND(00000018): Bind i/f Virtual-Template2
*Jul  7 03:28:31.547: ppp14 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:28:33.555: AAA/AUTHOR (0x18): Pick method list 'default'
*Jul  7 03:28:33.555: AAA/BIND(00000019): Bind i/f
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ, flg TLS, ver 2, len 141, tnl 0,
ns 0, nr 0
         C8 02 00 8D 00 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 10 00 00 00 07 4C 41 43 2D 74 75
         6E 6E 65 6C 00 19 00 00 00 08 43 69 73 63 6F 20
         53 79 73 74 65 6D 73 ...
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse SCCRP
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 2, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Protocol Ver 256
```

```
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 3, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Framing Cap 0x0
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 4, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Bearer Cap 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 6, len 8, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Firmware Ver 0x1120
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 7, len 16, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Hostname LNS-tunnel
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 8, len 25, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Vendor Name Cisco Systems, Inc.
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 9, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Assigned Tunnel ID 6897
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 10, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Rx Window Size 20050
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 11, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng
         81 13 03 F6 A8 E4 1D DD 25 18 25 6E 67 8C 7C 39
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 13, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng Resp
         4D 52 91 DC 1A 43 B3 31 B4 F5 B8 E1 88 22 4F 41
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: No missing AVPs in SCCRP
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP, flg TLS, ver 2, len 157, tnl
5192, ns 0, nr 1
contiguous pak, size 157
         C8 02 00 9D 14 48 00 00 00 00 00 01 80 08 00 00
         00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
         00 03 00 00 00 00 80 0A 00 00 00 04 00 00 00 00
         00 08 00 00 00 06 11 20 80 10 00 00 00 07 4C 4E
         53 2D 74 75 6E 6E 65 6C ...
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP from LNS-tunnel
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN  to LNS-tunnel tnlid 6897
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN, flg TLS, ver 2, len 42, tnl
6897, ns 1, nr 1
         C8 02 00 2A 1A F1 00 00 00 01 00 01 80 08 00 00
         00 00 00 03 80 16 00 00 00 0D 32 24 17 BC 6A 19
         B1 79 F3 F9 A9 D4 67 7D 9A DB
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ to LNS-tunnel 6897/0
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ, flg TLS, ver 2, len
63, tnl 6897, lsid 11, rsid 0, ns 2, nr 1
         C8 02 00 3F 1A F1 00 00 00 02 00 01 80 08 00 00
         00 00 00 0A 80 0A 00 00 00 0F C8 14 B4 03 80 08
         00 00 00 0E 00 0B 80 0A 00 00 00 12 00 00 00 00
         00 0F 00 09 00 64 0F 10 09 02 02 00 1B 00 00
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 0, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 14, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Assigned Call ID 5
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: No missing AVPs in ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: I ICRP, flg TLS, ver 2, len
28, tnl 5192, lsid 11, rsid 0, ns 1, nr 3
contiguous pak, size 28
         C8 02 00 1C 14 48 00 0B 00 01 00 03 80 08 00 00
         00 00 00 0B 80 08 00 00 00 0E 00 05
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN to LNS-tunnel 6897/5
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN, flg TLS, ver 2, len
167, tnl 6897, lsid 11, rsid 5, ns 3, nr 2
         C8 02 00 A7 1A F1 00 05 00 03 00 02 80 08 00 00
         00 00 00 0C 80 0A 00 00 00 18 06 1A 80 00 00 0A
         00 00 00 26 06 1A 80 00 80 0A 00 00 00 13 00 00
         00 01 00 15 00 00 00 1B 01 04 05 D4 03 05 C2 23
         05 05 06 0A 0B E2 7A ...
*Jul  7 03:28:33.579: RADIUS/ENCODE(00000018):Orig. component type = PPoE
```

```
*Jul  7 03:28:33.579: RADIUS(00000018): Config NAS IP: 10.0.0.0
*Jul  7 03:28:33.579: RADIUS(00000018): sending
*Jul  7 03:28:33.579: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:28:33.579: RADIUS(00000018): Send Accounting-Request to
172.19.192.238:2196 id 1646/23, len 176
*Jul  7 03:28:33.579: RADIUS:  authenticator 3C 81 D6 C5 2B 6D 21 8E - 19 FF
43 B5 41 86 A8 A5
*Jul  7 03:28:33.579: RADIUS:  Acct-Session-Id     [44]  10  "00000023"
*Jul  7 03:28:33.579: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:28:33.579: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                   [1]
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Assignment-Id[82]  5   "lac"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                   [3]
*Jul  7 03:28:33.583: RADIUS:  Acct-Tunnel-Connecti[68]  12  "3356800003"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Auth-I[90]  12  "LAC-tunnel"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Auth-I[91]  12  "LNS-tunnel"
*Jul  7 03:28:33.583: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:28:33.583: RADIUS:  Acct-Authentic      [45]  6
Local                     [2]
*Jul  7 03:28:33.583: RADIUS:  Acct-Status-Type    [40]  6
Start                     [1]
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:28:33.583: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:28:33.583: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:28:33.583: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:28:33.583: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:28:33.683: RADIUS: Received from id 1646/23 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:28:33.683: RADIUS:  authenticator 1C E9 53 42 A2 8A 58 9A - C3 CC
1D 79 9F A4 6F 3A
```

Additional References

The following sections provide references related to Per VRF AAA.

Related Documents

Related Topic	Document Title
Configuring server groups	Configuring RADIUS chapter in the *Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2.
RADIUS attribute screening	RADIUS Attribute Value Screening chapter in the *Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2.
Configuring broadcast accounting	Configuring Accounting chapter in the *Cisco IOS XE Security Configuration Guide: Securing User Services*, Release 2.

Related Topic	Document Title
Cisco IOS Security Commands	*Cisco IOS Security Command Reference*
Cisco IOS Switching Services Commands	*Cisco IOS IP Switching Command Reference*
Configuring Multiprotocol Label Switching	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*, Release 2
Configuring virtual templates	Virtual Templates and Profiles section of the *Cisco IOS XE Dial Technologies Configuration Guide*, Release 2

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Per VRF AAA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 30: Feature Information for Per VRF AAA

Feature Name	Releases	Feature Information
Per VRF AAA	Cisco IOS XE Release 2.1	The Per VRF AAA feature allows authentication, authorization, and accounting (AAA) on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa accounting, aaa accounting delay-start, ip radius source-interface, server-private (RADIUS)**, **ip vrf forwarding (server-group)**, **radius-server domain-stripping**, **aaa authorization template**.
RADIUS Per-VRF Server Group	Cisco IOS XE Release 2.1	Using the Radius Per-VRF Server Group feature, Internet Service Providers (ISPs) can partition RADIUS server groups based on Virtual Route Forwarding (VRF). This means that you can define RADIUS server groups that belong to a VRF. This feature is supported by “aaa: rad-serv-vrf” VSA. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ip vrf forwarding**.
Attribute Filtering Per-Domain and VRF Aware Framed-Routes	Cisco IOS XE Release 2.3	The Attribute Filtering Per-Domain and VRF Aware Framed-Routes feature allows for attribute filtering per-domain and VRF aware Framed-Routes. It introduces support for the “aaa:rad-serv-filter” VSA. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
AAA CLI Stop Record Enhancement	Cisco IOS XE Release 2.4	The AAA CLI Stop Record Enhancement feature enables sending an accounting stop record only when an access accept is received from the AAA server. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa accounting send stop-record authentication**.

Feature Name	Releases	Feature Information
Dynamic Per VRF AAA	Cisco IOS XE Release 2.4	The Dynamic Per VRF AAA feature allows you to use a customer template, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Glossary

AAA--authentication, authorization, and accounting. A framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

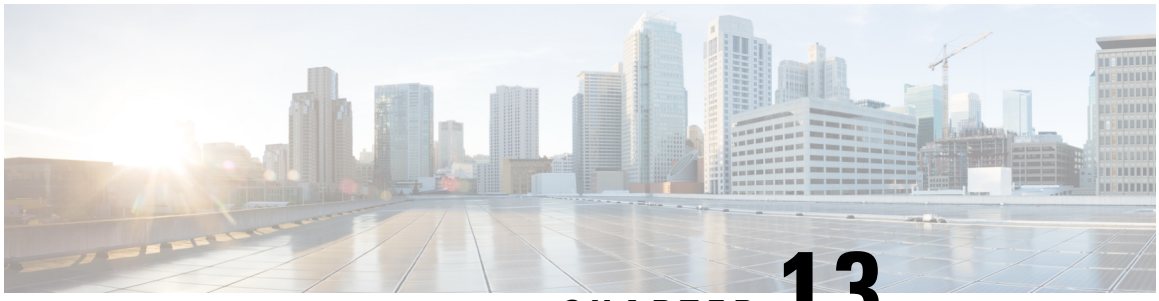
L2TP--Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

PE--Provider Edge. Networking devices that are located on the edge of a service provider network.

RADIUS --Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VPN --Virtual Private Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

VRF --Virtual Route Forwarding. Initially, a router has only one global default routing/forwarding table. VRFs can be viewed as multiple disjoined routing/forwarding tables, where the routes of a user have no correlation with the routes of another user.



CHAPTER 13

AAA Support for IPv6

Authentication, authorization, and accounting (AAA) support for IPv6 is in compliance with RFC 3162. This module provides information about how to configure AAA options for IPv6.

Information About AAA Support for IPv6

AAA over IPv6

Vendor-specific attributes (VSAs) are used to support Authentication, Authorization and Accounting(AAA) over IPv6. Cisco VSAs are inacl, outacl, prefix, and route.

You can configure prefix pools and pool names by using the AAA protocol. Customers can deploy an IPv6 RADIUS server or a TACACS+ server to communicate with Cisco devices.

AAA Support for IPv6 RADIUS Attributes

The following RADIUS attributes, as described in RFC 3162, are supported for IPv6:

- Framed-Interface-Id
- Framed-IPv6-Pool
- Framed-IPv6-Prefix
- Framed-IPv6-Route
- Login-IPv6-Host

The following RADIUS attributes are also supported for IPv6:

- Delegated-IPv6-Prefix (RFC 4818)
- Delegated-IPv6-Prefix-Pool

- DNS-Server-IPv6-Address
- IPv6 ACL
- IPv6_DNS_Servers
- IPv6 Pool
- IPv6 Prefix#
- IPv6 Route

The attributes listed above can be configured on a RADIUS server and downloaded to access servers, where they can be applied to access connections.

Prerequisites for Using AAA Attributes for IPv6

AAA attributes for IPv6 are compliant with RFC 3162 and require a RADIUS server capable of supporting RFC 3162.

RADIUS Per-User Attributes for Virtual Access in IPv6 Environments

The following IPv6 RADIUS attributes are supported for virtual access and can be used as attribute-value (AV) pairs:

- Delegated-IPv6-Prefix
- Delegated-IPv6-Prefix-Pool
- DNS-Server-IPv6-Address
- Framed-Interface-Id
- Framed-IPv6-Pool
- Framed-IPv6-Prefix
- Framed-IPv6-Route
- IPv6 ACL
- IPv6_DNS_Servers
- IPv6 Pool
- IPv6 Prefix#
- IPv6 Route
- Login-IPv6-Host

Delegated-IPv6-Prefix

The Delegated-IPv6-Prefix attribute indicates an IPv6 prefix to be delegated to a user for use in a network. This attribute is used during DHCP prefix delegation between a RADIUS server and a delegating device. A Network Access Server (NAS) that hosts a DHCP Version 6 (DHCPv6) server can act as a delegating device.

The following example shows how to use the Delegated-IPv6-Prefix attribute:

```
ipv6:delegated-prefix=2001:DB8::/64
```

Note The Cisco VSA format is not supported for this attribute. If you try to add this attribute in the Cisco VSA format into a user profile, the RADIUS server response fails. Use only the IETF attribute format for this attribute.

Delegated-IPv6-Prefix-Pool

The Delegated-IPv6-Prefix-Pool attribute indicates the name of a prefix pool from which a prefix is selected and delegated to a device.

Prefix delegation is a DHCPv6 option for delegating IPv6 prefixes. Prefix delegation involves a delegating device that selects a prefix and assigns it on a temporary basis to a requesting device. A delegating device uses many strategies to choose a prefix. One method is to choose a prefix from a prefix pool with a name that is defined locally on a device.

The Delegated-IPv6-Prefix-Pool attribute indicates the name of an assigned prefix pool. A RADIUS server uses this attribute to communicate the name of a prefix pool to a NAS hosting a DHCPv6 server and acting as a delegating device.

You may use DHCPv6 prefix delegation along with ICMPv6 stateless address autoconfiguration (SLAAC) on a network. In this case, both the Delegated-IPv6-Prefix-Pool attribute and the Framed-IPv6-Pool attribute may be included within the same packet. To avoid ambiguity, the Delegated-IPv6-Prefix-Pool attribute should be restricted to the authorization and accounting of prefix pools used in DHCPv6 delegation, and the Framed-IPv6-Pool attribute should be used for the authorization and accounting of prefix pools used in SLAAC.

The following example shows how an address prefix is selected from a pool named pool1. The prefix pool pool1 is downloaded to a delegating device from a RADIUS server by using the Delegated-IPv6-Prefix-Pool attribute. The device then selects the address prefix 2001:DB8::/64 from this prefix pool.

```
Cisco:Cisco-AVpair = "ipv6:delegated-ipv6-pool = pool1"
!
ipv6 dhcp pool pool1
address prefix 2001:DB8::/64
!
```

DNS-Server-IPv6-Address

The DNS-Server-IPv6-Address attribute indicates the IPv6 address of a Domain Name System (DNS) server. A DHCPv6 server can configure a host with the IPv6 address of a DNS server. The IPv6 address of the DNS server can also be conveyed to the host using router advertisement messages from ICMPv6 devices.

A NAS may host a DHCPv6 server to handle DHCPv6 requests from hosts. The NAS may also act as a device that provides router advertisement messages. Therefore, this attribute is used to provide the NAS with the IPv6 address of the DNS server.

If a NAS has to announce more than one recursive DNS server to a host, this attribute can be included multiple times in Access-Accept packets sent from the NAS to the host.

The following example shows how you can define the IPv6 address of a DNS server by using the DNS-Server-IPv6-Address attribute:

```
Cisco:Cisco-AVpair = "ipv6:ipv6-dns-servers-addr=2001:DB8::"
```

Framed-Interface-Id

The Framed-Interface-Id attribute indicates an IPv6 interface identifier to be configured for a user.

This attribute is used during IPv6 Control Protocol (IPv6CP) negotiations of the Interface-Identifier option. If negotiations are successful, the NAS uses this attribute to communicate a preferred IPv6 interface identifier to the RADIUS server by using Access-Request packets. This attribute may also be used in Access-Accept packets.

Framed-IPv6-Pool

The Framed-IPv6-Pool attribute indicates the name of a pool that is used to assign an IPv6 prefix to a user. This pool should be either defined locally on a device or defined on a RADIUS server from where pools can be downloaded.

Framed-IPv6-Prefix

The Framed-IPv6-Prefix attribute indicates an IPv6 prefix (and a corresponding route) to be configured for a user. So this attribute performs the same function as a Cisco VSA and is used for virtual access only. A NAS uses this attribute to communicate a preferred IPv6 prefix to a RADIUS server by using Access-Request packets. This attribute may also be used in Access-Accept packets and can appear multiple times in these packets. The NAS creates a corresponding route for the prefix.

This attribute is used by a user to specify which prefixes to advertise in router advertisement messages of the Neighbor Discovery Protocol.

This attribute can also be used for DHCPv6 prefix delegation, and a separate profile must be created for a user on the RADIUS server. The username associated with this separate profile has the suffix “-dhcpv6”.

The Framed-IPv6-Prefix attribute is treated differently in this separate profile and the regular profile of a user. If a NAS needs to send a prefix through router advertisement messages, the prefix is placed in the Framed-IPv6-Prefix attribute of the regular profile of the user. If a NAS needs to delegate a prefix to the network of a remote user, the prefix is placed in the Framed-IPv6-Prefix attribute of the separate profile of the user.

Note The RADIUS IETF attribute format and the Cisco VSA format are supported for this attribute.

Framed-IPv6-Route

The Framed-IPv6-Route attribute indicates the routing information to be configured for a user on a NAS. This attribute performs the same function as a Cisco VSA. The value of the attribute is a string and is specified by using the **ipv6 route** command.

IPv6 ACL

The IPv6 ACL attribute is used to specify a complete IPv6 access list. The unique name of an access list is generated automatically. An access list is removed when the respective user logs out. The previous access list on the interface is then reapplied.

The inacl and outacl attributes enable you to specify an existing access list configured on a device. The following example shows how to define an access list identified with number 1:

```
cisco-avpair = "ipv6:inacl#1=permit 2001:DB8:cc00:1::/48",
cisco-avpair = "ipv6:outacl#1=deny 2001:DB8::/10",
```

IPv6_DNS_Servers

The IPv6_DNS_Servers attribute is used to send up to two DNS server addresses to the DHCPv6 server. The DNS server addresses are saved in the interface DHCPv6 subblock and override other configurations in the DHCPv6 pool. This attribute is also included in attributes returned for AAA start and stop notifications.

IPv6 Pool

The IPv6 Pool attribute extends the IPv4 address pool attribute to support the IPv6 protocol for RADIUS authentication. This attribute specifies the name of a local pool on a NAS from which a prefix is chosen and used whenever PPP is configured and the protocol is specified as IPv6. The address pool works with local pooling and specifies the name of a local pool that is preconfigured on the NAS.

IPv6 Prefix#

The IPv6 Prefix# attribute indicates which prefixes to advertise in router advertisement messages of the Neighbor Discovery Protocol. When this attribute is used, a corresponding route (marked as a per-user static route) is installed in the routing information base (RIB) tables for a given prefix.

The following example shows how to specify which prefixes to advertise:

```
cisco-avpair = "ipv6:prefix#1=2001:DB8::/64",
cisco-avpair = "ipv6:prefix#2=2001:DB8::/64",
```

IPv6 Route

The IPv6 Route attribute is used to specify a static route for a user. A static route is appropriate when Cisco software cannot dynamically build a route to the destination. See the **ipv6 route** command for more information about building static routes.

The following example shows how to use the IPv6 Route attribute to define a static route:

```
cisco-avpair = "ipv6:route#1=2001:DB8:cc00:1::/48",
cisco-avpair = "ipv6:route#2=2001:DB8:cc00:2::/48",
```

Login-IPv6-Host

The Login-IPv6-Host attribute indicates IPv6 addresses of hosts with which to connect a user when the Login-Service attribute is included. A NAS uses the Login-IPv6-Host attribute in Access-Request packets to communicate to a RADIUS server that it prefers to use certain hosts.

How to Configure AAA Support for IPv6

Configuring DHCPv6 AAA Options

Perform the following task to configure the option of acquiring prefixes from the AAA server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *pool-name*
4. **prefix-delegation aaa** [**method-list** *method-list*] [*lifetime*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *pool-name* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters IPv6 DHCP pool configuration mode.
Step 4	**prefix-delegation aaa** [**method-list** *method-list*] [*lifetime*] **Example:** `Device(config-dhcpv6)# prefix-delegation aaa method-list list1`	Specifies that prefixes are to be acquired from AAA servers.
Step 5	**end** **Example:** `Device(config-dhcpv6)# end`	Exits IPv6 DHCP pool configuration mode and returns to privileged EXEC mode.

Configuration Examples for AAA Support for IPv6

Example: DHCPv6 AAA Options Configuration

The following example shows how to configure the DHCPv6 option of acquiring prefixes from the AAA server:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 dhcp pool pool1
Device(config-dhcpv6)# prefix-delegation aaa method-list list1
Device(config-dhcpv6)# end
```

Example: RADIUS Configuration

The following sample RADIUS configuration shows the definition of AV pairs to establish static routes:

```
campus1 Auth-Type = Local, Password = "mypassword"
                User-Service-Type = Framed-User,
                Framed-Protocol = PPP,
                cisco-avpair = "ipv6:inacl#1=permit 2001:DB8:1::/64 any",
                cisco-avpair = "ipv6:route=2001:DB8:2::/64",
                cisco-avpair = "ipv6:route=2001:DB8:3::/64",
                cisco-avpair = "ipv6:prefix=2001:DB8:2::/64 0 0 onlink autoconfig",
                cisco-avpair = "ipv6:prefix=2001:DB8:3::/64 0 0 onlink autoconfig",
                cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
```

Additional References

Related Documents

Related Topic	Document Title
Using the **time-range** command to establish time ranges	The chapter *Performing Basic System Management* in the *Cisco IOS XE Network Management Configuration Guide*
Network management command descriptions	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

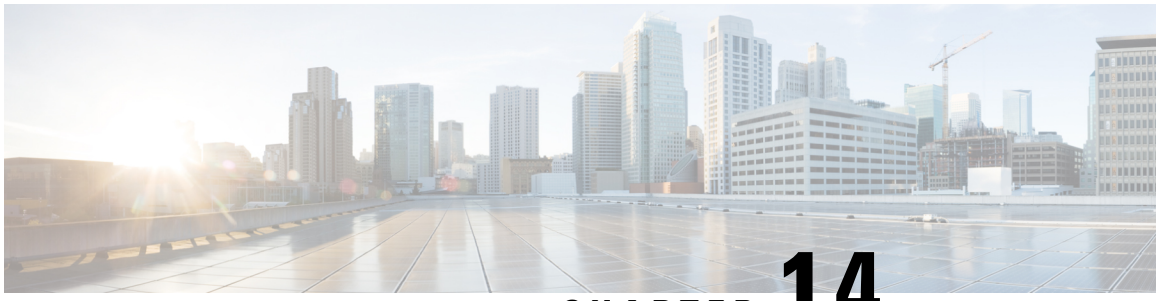
Feature Information for RADIUS over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 31: Feature Information for RADIUS over IPv6

Feature Name	Releases	Feature Information
RADIUS over IPv6	15.1(1)SY	RADIUS attributes defined in RFC 3162 are supported.



CHAPTER 14

TACACS+ over IPv6

An IPv6 server can be configured to be used with TACACS+.

Information About TACACS+ over IPv6

The Terminal Access Controller Access-Control System (TACACS+) security protocol provides centralized validation of users. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your devices are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service (authentication, authorization, and accounting) independently. Each service is associated with its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

AAA over IPv6

Vendor-specific attributes (VSAs) are used to support Authentication, Authorization and Accounting(AAA) over IPv6. Cisco VSAs are inacl, outacl, prefix, and route.

You can configure prefix pools and pool names by using the AAA protocol. Customers can deploy an IPv6 RADIUS server or a TACACS+ server to communicate with Cisco devices.

TACACS+ Over an IPv6 Transport

An IPv6 server can be configured to use TACACS+. Both IPv6 and IPv4 servers can be configured to use TACACS+ using a name instead of an IPv4 or IPv6 address.

How to Configure TACACS+ over IPv6

Configuring the TACACS+ Server over IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **tacacs server** *name*
4. **address ipv6** *ipv6-address*
5. **key** [**0** | **7**] *key-string*
6. **port** [*number*
7. **send-nat-address**
8. **single-connection**
9. **timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**tacacs server** *name* **Example:** `Device(config)# tacacs server server1`	Configures the TACACS+ server for IPv6 and enters TACACS+ server configuration mode.
Step 4	**address ipv6** *ipv6-address* **Example:** `Device(config-server-tacacs)# address ipv6 2001:DB8:3333:4::5`	Configures the IPv6 address of the TACACS+ server.
Step 5	**key** [**0**	**7**] *key-string* **Example:** `Device(config-server-tacacs)# key 0 key1`	Configures the per-server encryption key on the TACACS+ server.

	Command or Action	Purpose
Step 6	**port** [*number* **Example:** `Device(config-server-tacacs)# port 12`	Specifies the TCP port to be used for TACACS+ connections.
Step 7	**send-nat-address** **Example:** `Device(config-server-tacacs)# send-nat-address`	Sends a client's post-NAT address to the TACACS+ server.
Step 8	**single-connection** **Example:** `Device(config-server-tacacs)# single-connection`	Enables all TACACS packets to be sent to the same server using a single TCP connection.
Step 9	**timeout** *seconds* **Example:** `Device(config-server-tacacs)# timeout 10`	Configures the time to wait for a reply from the specified TACACS server.

Specifying the Source Address in TACACS+ Packets

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 tacacs source-interface** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 tacacs source-interface** *type number* **Example:**	Specifies an interface to use for the source address in TACACS+ packets.

	Command or Action	Purpose
	`Router(config)# ipv6 tacacs source-interface GigabitEthernet 0/0/0`	

Configuring TACACS+ Server Group Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server tacacs+** *group-name*
4. **server name** *server-name*
5. **server-private** {*ip-address* | *name* | *ipv6-address*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server tacacs+** *group-name* **Example:** `Device(config)# aaa group server tacacs+ group1`	Groups different TACACS+ server hosts into distinct lists and distinct methods.
Step 4	**server name** *server-name* **Example:** `Device(config-sg-tacacs+)# server name server1`	Specifies an IPv6 TACACS+ server.
Step 5	**server-private** {*ip-address* | *name* | *ipv6-address*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Device(config-sg-tacacs+)# server-private 2001:DB8:3333:4::5 port 19 key key1`	Configures the IPv6 address of the private TACACS+ server for the group server.

Configuration Examples for TACACS+ over IPv6

Example: Configuring TACACS+ Server over IPv6

```
Device# show tacacs

           Tacacs+ Server:          server1
           Server Address:          FE80::200:F8FF:FE21:67CF
             Socket opens:          0
            Socket closes:          0
            Socket aborts:          0
            Socket errors:          0
         Socket Timeouts:           0
  Failed Connect Attempts:          0
       Total Packets Sent:          0
       Total Packets Recv:          0
```

Additional References

The following sections provide references related to the MSCHAP Version 2 feature.

Related Documents

Related Topic	Document Title
Configuring PPP interfaces	PPP Configuration in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T.
Descriptions of the tasks and commands necessary to configure and maintain Cisco networking devices	*Cisco IOS Dial Technologies Command Reference*
Lists of IOS Security Commands	*Cisco IOS Security Command Reference*
Configuring PPP authentication using AAA	Configuring PPP Authentication Using AAA in the Configuring Authentication module in the *Cisco IOS Security Configuration Guide: Securing User Services* , Release 12.4T.
Configuring RADIUS Authentication	Configuring RADIUS module in the *Cisco IOS Security Configuration Guide: Securing User Services*, Release 12.4T.

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1661	*Point-to-Point Protocol (PPP)*
RFC 2548	*Microsoft Vendor-specific RADIUS Attributes*
RFC 2759	*Microsoft PPP CHAP Extensions, Version 2*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

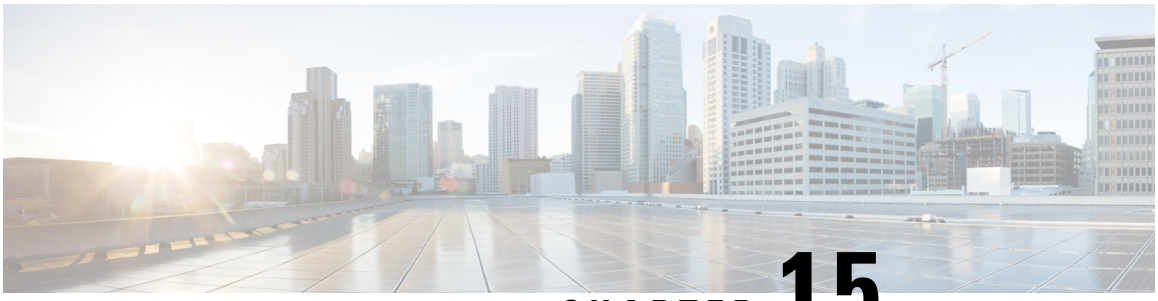
Feature Information for TACACS+ over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 32: Feature Information for TACACS+ over IPv6

Feature Name	Releases	Feature Information
TACACS+ over IPv6	Cisco IOS XE Release 3.2S	TACACS+ over IPv6 is supported. The following commands were introduced or modified: **aaa group server tacacs+**, **address ipv6 (TACACS+)**, **ipv6 tacacs source-interface**, **key (TACACS+)**, **port (TACACS+)**, **send-nat-address**, **server name (IPv6 TACACS+)**, **server-private (TACACS+)**, **single-connection**, **tacacs server**, **timeout (TACACS+)**.



CHAPTER 15

AAA Dead-Server Detection

The AAA Dead-Server Detection feature allows you to configure the criteria to be used to mark a RADIUS server as dead. If no criteria are explicitly configured, the criteria are computed dynamically on the basis of the number of outstanding transactions. Using this feature will result in less deadtime and quicker packet processing.

Prerequisites for AAA Dead-Server Detection

- You must have access to a RADIUS server.
- You should be familiar with configuring a RADIUS server.
- You should be familiar with configuring authentication, authorization, and accounting (AAA).
- Before a server can be marked as dead, you must first configure the **radius-server deadtime** command. If this command is not configured, even if the criteria are met for the server to be marked as dead, the server state will be the “up” state.

Restrictions for AAA Dead-Server Detection

- Original transmissions are not counted in the number of consecutive timeouts that must occur on the router before the server is marked as dead--only the number of retransmissions are counted.

Information About AAA Dead-Server Detection

Criteria for Marking a RADIUS Server As Dead

The AAA Dead-Server Detection feature allows you to determine the criteria that are used to mark a RADIUS server as dead. That is, you can configure the minimum amount of time, in seconds, that must elapse from the time that the router last received a valid packet from the RADIUS server to the time the server is marked as dead. If a packet has not been received since the router booted, and there is a timeout, the time criterion will be treated as though it has been met.

In addition, you can configure the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. If the server performs both authentication and accounting, both types of packets are included in the number. Improperly constructed packets are counted as though they are timeouts. Only retransmissions are counted, not the initial transmission. (Each timeout causes one retransmission to be sent.)

Note Both the time criterion and the tries criterion must be met for the server to be marked as dead.

The RADIUS dead-server detection configuration will result in the prompt detection of RADIUS servers that have stopped responding. This configuration will also result in the avoidance of servers being improperly marked as dead when they are "swamped" (responding slowly) and the avoidance of the state of servers being rapidly changed from dead to live to dead again. This prompt detection of nonresponding RADIUS servers and the avoidance of swamped and dead-to-live-to-dead-again servers will result in less deadtime and quicker packet processing.

How to Configure AAA Dead-Server Detection

Configuring AAA Dead-Server Detection

To configure AAA Dead-Server Detection, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server deadtime** *minutes*
5. **radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**radius-server deadtime** *minutes* **Example:** `Router (config)# radius-server deadtime 5`	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 5	**radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*] **Example:** `Router (config)# radius-server dead-criteria time 5 tries 4`	Forces one or both of the criteria--used to mark a RADIUS server as dead--to be the indicated constant.

Troubleshooting Tips

After you have configured AAA Dead-Server Detection, you should verify your configuration using the **show running-config** command. This verification is especially important if you have used the **no** form of the **radius-server dead-criteria** command. The output of the **show running-config** command must show the same values in the "Dead Criteria Details" field that you configured using the **radius-server dead-criteria** command.

Verifying AAA Dead-Server Detection

To verify your AAA Dead-Server Detection configuration, perform the following steps. The **show** and **debug** commands may be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug aaa dead-criteria transactions**
3. **show aaa dead-criteria**
4. **show aaa servers** [**private** | **public**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa dead-criteria transactions** **Example:** `Router# debug aaa dead-criteria transactions`	Displays AAA dead-criteria transaction values.
Step 3	**show aaa dead-criteria** **Example:** `Router# show aaa dead-criteria`	Displays dead-criteria information for a AAA server.
Step 4	**show aaa servers** [**private** | **public**] **Example:** `Router# show aaa server private`	Displays the status and number of packets that are sent to and received from all public and private authentication, authorization, and accounting (AAA) RADIUS servers. • The **private** keyword optionally displays the AAA servers only. • The **public** keyword optionally displays the AAA servers only.

Configuration Examples for AAA Dead-Server Detection

Configuring AAA Dead-Server Detection Example

The following example shows that the router will be considered dead after 5 seconds and four tries:

```
Router (config)# aaa new-model
Router (config)# radius-server deadtime 5
Router (config)# radius-server dead-criteria time 5 tries 4
```

debug aaa dead-criteria transactions Command Example

The following output example shows dead-criteria transaction information for a particular server group:

```
Router# debug aaa dead-criteria transactions
AAA Transaction debugs debugging is on
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Retransmit Tries: 22, Current Max Tries: 22
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Dead Detect Interval: 25s, Current Max
Interval: 25s
```

```
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Estimated Outstanding Transactions: 6, Current Max
Transactions: 6
```

show aaa dead-criteria Command Example

The following output example shows that dead-server-detection information has been requested for a RADIUS server at the IP address 172.19.192.80:

```
Router# show aaa dead-criteria radius 172.19.192.80 radius
RADIUS Server Dead Criteria:
============================
Server Details:
    Address : 172.19.192.80
    Auth Port : 1645
    Acct Port : 1646
Server Group : radius
Dead Criteria Details:
    Configured Retransmits : 62
    Configured Timeout : 27
    Estimated Outstanding Transactions: 5
    Dead Detect Time : 25s
    Computed Retransmit Tries: 22
    Statistics Gathered Since Last Successful Transaction
=====================================================
Max Computed Outstanding Transactions: 5
Max Computed Dead Detect Time: 25s
Max Computed Retransmits : 22
```

Additional References

The following sections provide references related to the AAA Dead-Server Detection feature.

Related Documents

Related Topic	Document Title
Configuring RADIUS	Configuring RADIUS feature module.
Configuring AAA	Configuring Authentication
	Configuring Authorization
	Configuring Accounting
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2865	*Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

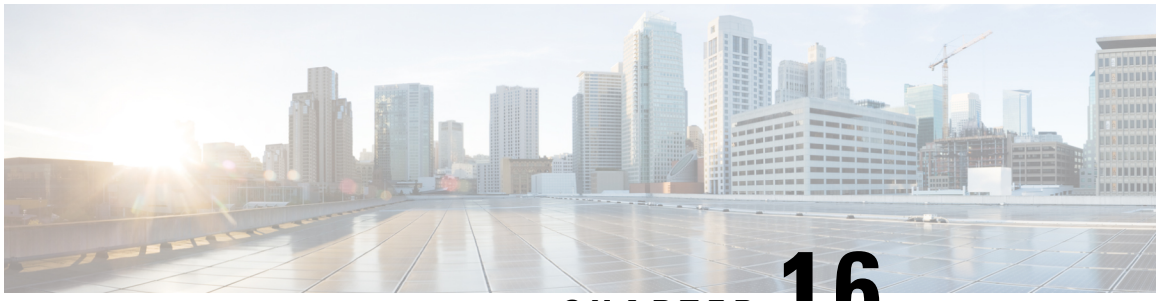
Feature Information for AAA Dead-Server Detection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 33: Feature Information for AAA Dead-Server Detection

Feature Name	Releases	Feature Information
AAA Dead-Server Detection	Cisco IOS XE Release 3.9S	Allows you to configure the criteria to be used to mark a RADIUS server as dead. The following commands were introduced or modified: **debug aaa dead-criteria transactions**, **radius-server dead-criteria**, **show aaa dead-criteria**, **show aaa servers**.



CHAPTER 16

Login Password Retry Lockout

The Login Password Retry Lockout feature allows system administrators to lock out a local authentication, authorization, and accounting (AAA) user account after a configured number of unsuccessful attempts by the user to log in.

Prerequisites for Login Password Retry Lockout

- You must be running a Cisco IOS image that contains the AAA component.

Restrictions for Login Password Retry Lockout

- Authorized users can lock themselves out because there is no distinction between an attacker who is guessing passwords and an authorized user who is entering the password incorrectly multiple times.
- A denial of service (DoS) attack is possible; that is, an authorized user could be locked out by an attacker if the username of the authorized user is known to the attacker.

Information About Login Password Retry Lockout

Lock Out of a Local AAA User Account

The Login Password Retry Lockout feature allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts by the user to log in using the username that corresponds

to the AAA user account. A locked-out user cannot successfully log in again until the user account is unlocked by the administrator.

A system message is generated when a user is either locked by the system or unlocked by the system administrator. The following is an example of such a system message:

```
%AAA-5-USER_LOCKED: User user1 locked out on authentication failure.
```

The system administrator cannot be locked out.

Note The system administrator is a special user who has been configured using the maximum privilege level (root privilege--level 15). A user who has been configured using a lesser privilege level can change the privilege level using the **enable** command. A user that can change to the root privilege (level 15) is able to act as a system administrator.

This feature is applicable to any login authentication method, such as ASCII, Challenge Handshake Authentication Protocol (CHAP), and Password Authentication Protocol (PAP).

Note No messages are displayed to users after authentication failures that are due to the locked status (that is, there is no distinction between a normal authentication failure and an authentication failure due to the locked status of the user).

How to Configure Login Password Retry Lockout

Configuring Login Password Retry Lockout

To configure the Login Password Retry Lockout feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **username** *name* [**privilege** *level*] **password** *encryption-type password*
4. **aaa new-model**
5. **aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts*
6. **aaa authentication login default method**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**username** *name* [**privilege** *level*] **password** *encryption-type password* **Example:** `Device(config)# username user1 privilege 15 password 0 cisco`	Establishes a username-based authentication system.
Step 4	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model.
Step 5	**aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts* **Example:** `Device(config)# aaa local authentication attempts max-fail 3`	Specifies the maximum number of unsuccessful attempts before a user is locked out.
Step 6	**aaa authentication login default method** **Example:** `Device(config)# aaa authentication login default local`	Sets the authentication, authorization, and accounting (AAA) authentication method at login. For example, **aaa authentication login default** *local* specifies the local AAA user database.

Unlocking a Login Locked-Out User

To unlock a login locked-out user, perform the following steps.

Note This task can be performed only by users having the root privilege (level 15).

SUMMARY STEPS

1. **enable**
2. **clear aaa local user lockout** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**clear aaa local user lockout** {**username** *username* | **all**} **Example:** `Device# clear aaa local user lockout username user1`	Unlocks a locked-out user.

Clearing the Unsuccessful Login Attempts of a User

This task is useful for cases in which the user configuration was changed and the unsuccessful login attempts of a user that are already logged must be cleared.

To clear the unsuccessful login attempts of a user that have already been logged, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear aaa local user fail-attempts** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear aaa local user fail-attempts** {**username** *username* | **all**} **Example:** `Device# clear aaa local user fail-attempts username user1`	Clears the unsuccessful attempts of the user. • This command is useful for cases in which the user configuration was changed and the unsuccessful attempts that are already logged must be cleared.

Monitoring and Maintaining Login Password Retry Lockout Status

To monitor and maintain the status of the Login Password Retry Lockout configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show aaa local user lockout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show aaa local user lockout** **Example:** `Device# show aaa local user lockout`	Displays a list of the locked-out users for the current login password retry lockout configuration.

Example

The following output shows that user1 is locked out:

```
Device# show aaa local user lockout
          Local-user          Lock time
          user1               04:28:49 UTC Sat Jun 19 2004
```

Configuration Examples for Login Password Retry Lockout

Displaying the Login Password Retry Lockout Configuration Example

The following **show running-config** command output illustrates that the maximum number of failed user attempts has been set for 2 as the login password retry lockout configuration:

```
Device # show running-config
Building configuration...
Current configuration : 1214 bytes
!
version 12.3
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname LAC-2
!
boot-start-marker
boot-end-marker
!
!
username sysadmin
username sysad privilege 15 password 0 cisco
username user1 password 0 cisco
aaa new-model
aaa local authentication attempts max-fail 2
!
!
```

```
aaa authentication login default local
aaa dnis map enable
aaa session-id common
```

Additional References

The following sections provide references related to Login Password Retry Lockout.

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Login Password Retry Lockout

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

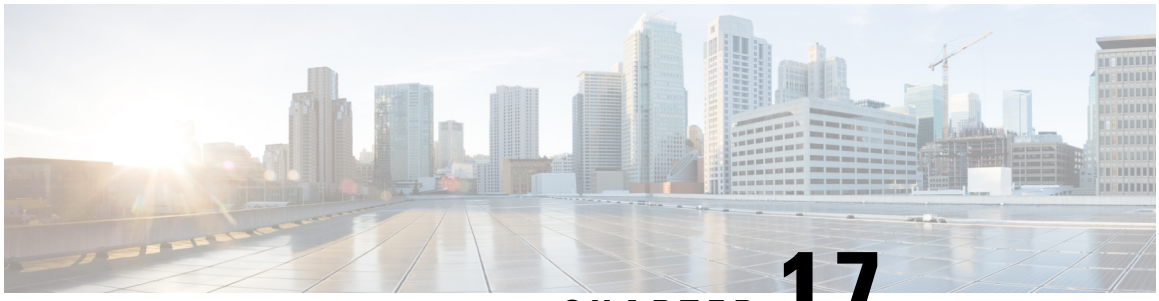
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 34: Feature Information for Login Password Retry Lockout

Feature Name	Releases	Feature Information
Login Password Retry Lockout	Cisco IOS XE Release 3.9S	The Login Password Retry Lockout feature allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts by the user to log in. The following commands were introduced or modified: **aaa local authentication attempts max-fail**, **clear aaa local user fail-attempts**, **clear aaa local user lockout**.

Glossary

- **local AAA method** --Method by which it is possible to configure a local user database on a router and to have AAA provision authentication or authorization of users from this database.
- **local AAA user** --User who is authenticated using the local AAA method.



CHAPTER 17

MSCHAP Version 2

The MSCHAP Version 2 feature (introduced in Cisco IOS Release 12.2(2)XB5) allows Cisco routers to utilize Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAP V2) authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server (NAS).

For Cisco IOS Release 12.4(6)T, MSCHAP V2 now supports a new feature: AAA Support for MSCHAPv2 Password Aging. Prior to Cisco IOS Release 12.4(6)T, when Password Authentication Protocol (PAP)-based clients sent username and password values to the authentication, authorization, and accounting (AAA) subsystem, AAA generated an authentication request to the RADIUS server. If the password expired, the RADIUS server replied with an authentication failure message. The reason for the authentication failure was not passed back to AAA subsystem; thus, users were denied access because of authentication failure but were not informed why they were denied access.

The Password Aging feature, available in Cisco IOS Release 12.4(6)T, notifies crypto-based clients that the password has expired and provides a generic way for the user to change the password. The Password Aging feature supports only crypto-based clients.

Prerequisites for MSCHAP Version 2

- Configure an interface type and enter interface configuration mode by using the **interface** command.
- Configure the interface for PPP encapsulation by using the **encapsulation** command.
- Be sure that the client operating system supports all MSCHAP V2 capabilities.
- For Cisco IOS Release 12.4(6)T, the Password Aging feature only supports RADIUS authentication for crypto-based clients.

- To ensure that the MSCHAP Version 2 features correctly interpret the authentication failure attributes sent by the RADIUS server, you must configure the **ppp max-bad-auth** command and set the number of authentication retries at two or more.

In addition, the **radius server vsa send authentication** command must be configured, allowing the RADIUS client to send a vendor-specific attribute to the RADIUS server. The Change Password feature is supported only for RADIUS authentication.

- The Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows NT operating systems have a known caveat that prevents the Change Password feature from working. You must download a patch from Microsoft at the following URL:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q326770

For more information on completing these tasks, see the section "PPP Configuration " in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T. The RADIUS server must be configured for authentication. Refer to vendor-specific documentation for information on configuring RADIUS authentication on the RADIUS server.

Restrictions for MSCHAP Version 2

- MSCHAP V2 authentication is not compatible with MSCHAP V1 authentication.
- The change password option is supported only for RADIUS authentication and is not available for local authentication.

Information About MSCHAP Version 2

MSCHAP V2 authentication is the default authentication method used by the Microsoft Windows 2000 operating system. Cisco routers that support this authentication method enable Microsoft Windows 2000 operating system users to establish remote PPP sessions without configuring an authentication method on the client.

MSCHAP V2 authentication introduced an additional feature not available with MSCHAP V1 or standard CHAP authentication: the Change Password feature. This features allows the client to change the account password if the RADIUS server reports that the password has expired.

Note MSCHAP V2 authentication is an updated version of MSCHAP that is similar to but incompatible with MSCHAP Version 1 (V1). MSCHAP V2 introduces mutual authentication between peers and a Change Password feature.

How to Configure MSCHAP Version 2

Configuring MSCHAP V2 Authentication

To configure the NAS to accept MSCHAP V2 authentication for local or RADIUS authentication and to allow proper interpretation of authentication failure attributes and vendor-specific RADIUS attributes for RADIUS authentication, use the following commands beginning in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send authentication**
4. **interface** *type number*
5. **ppp max-bad-auth** *number*
6. **ppp authentication ms-chap-v2**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server vsa send authentication** **Example:** `Device(config)# radius-server vsa send authentication`	Configures the NAS to recognize and use vendor-specific attributes.
Step 4	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/1`	Configures an interface type and enters interface configuration mode.
Step 5	**ppp max-bad-auth** *number* **Example:** `Device(config-if)# ppp max-bad-auth 2`	Configures a point-to-point interface to reset immediately after an authentication failure or within a specified number of authentication retries.

	Command or Action	Purpose
		• The default value for the *number*argument is 0 seconds (immediately). • The range is between 0 and 255. **Note** The *number* argument must be set to a value of at least 2 for authentication failure attributes to be interpreted by the NAS.
Step 6	**ppp authentication ms-chap-v2** **Example:** `Device(config-if)# ppp authentication ms-chap-v2`	Enables MSCHAP V2 authentication on a NAS.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying MSCHAP V2 Configuration

To verify that the MSCHAP Version 2 feature is configured properly, perform the following steps.

SUMMARY STEPS

1. **show running-config interface** *type number*
2. **debug ppp negotiation**
3. **debug ppp authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show running-config interface** *type number* **Example:** `Device# show running-config interface Asynch65`	Verifies the configuration of MSCHAP V2 as the authentication method for the specified interface.
Step 2	**debug ppp negotiation** **Example:** `Device# debug ppp negotiation`	Verifies successful MSCHAP V2 negotiation.
Step 3	**debug ppp authentication** **Example:** `Device# debug ppp authentication`	Verifies successful MSCHAP V2 authentication.

Configuring Password Aging for Crypto-Based Clients

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login**command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login**command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

After the RADIUS server requests a new password, AAA queries the crypto client, which in turn prompts the user to enter a new password.

To configure login authentication and password aging for crypto-based clients, use the following commands beginning in global configuration mode.

Note The AAA Password Expiry infrastructure notifies the Easy VPN client that the password has expired and provides a generic way for the user to change the password. Please use RADIUS-server domain-stripping feature wisely in combination with AAA password expiry support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*]
5. **crypto map** *map-name* **client authentication list** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa authentication login** {**default** | *list-name*} **passwd-expiry** *method1* [*method2...*] **Example:**	Enables password aging for crypto-based clients on a local authentication list.

	Command or Action	Purpose
	`Device(config)# aaa authentication login userauthen passwd-expiry group radius`	
Step 5	**crypto map** *map-name* **client authentication list** *list-name* **Example:** **Example:** `Device(config)# crypto map clientmap client authentication list userauthen`	Configures user authentication (a list of authentication methods) on an existing crypto map.

Configuration Examples

Configuring Local Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication locally:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  username client password secret
```

Configuring RADIUS Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication via RADIUS:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  exit
aaa authentication ppp default group radius
 radius-server host 10.0.0.2 255.0.0.0
 radius-server key secret
 radius-server vsa send authentication
```

Configuring Password Aging with Crypto Authentication Example

The following example configures password aging by using AAA with a crypto-based client:

```
aaa authentication login userauthen passwd-expiry group radius
!
aaa session-id common
!
crypto isakmp policy 3
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group 3000client
 key cisco123
 dns 10.1.1.10
 wins 10.1.1.20
 domain cisco.com
 pool ippool
 acl 153
!
crypto ipsec transform-set myset esp-3des esp-sha-hmac
!
crypto dynamic-map dynmap 10
 set transform-set myset
!
crypto map clientmap client authentication list userauthen
!
radius-server host 10.140.15.203 auth-port 1645 acct-port 1646
radius-server domain-stripping prefix-delimiter $
radius-server key cisco123
radius-server vsa send authentication
radius-server vsa send authentication 3gpp2
!
end
```

Additional References

The following sections provide references related to the MSCHAP Version 2 feature.

Related Documents

Related Topic	Document Title
Configuring PPP interfaces	PPP Configuration in the *Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T.
Descriptions of the tasks and commands necessary to configure and maintain Cisco networking devices	*Cisco IOS Dial Technologies Command Reference*
Lists of IOS Security Commands	*Cisco IOS Security Command Reference*
Configuring PPP authentication using AAA	Configuring PPP Authentication Using AAA in the Configuring Authentication module in the *Cisco IOS Security Configuration Guide: Securing User Services* , Release 12.4T.

Related Topic	Document Title
Configuring RADIUS Authentication	Configuring RADIUS module in the *Cisco IOS Security Configuration Guide: Securing User Services*, Release 12.4T.

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1661	*Point-to-Point Protocol (PPP)*
RFC 2548	*Microsoft Vendor-specific RADIUS Attributes*
RFC 2759	*Microsoft PPP CHAP Extensions, Version 2*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

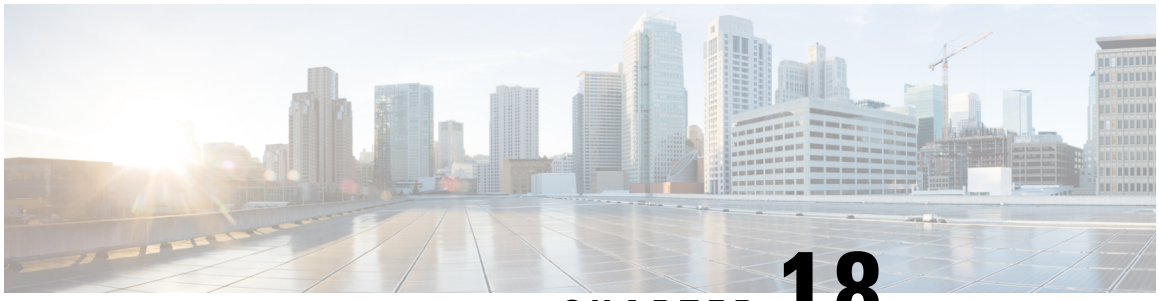
Feature Information for MSCHAP Version 2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for MSCHAP Version 2

Feature Name	Releases	Feature Information
MSCHAP Version 2	Cisco IOS XE Release 3.9S	The MSCHAP Version 2 feature allows Cisco routers to utilize Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAP V2) authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server (NAS). The following commands were introduced or modified: **aaa authentication login**, and **ppp authentication ms-chap-v2.**



CHAPTER 18

AAA Broadcast Accounting-Mandatory Response Support

The AAA Broadcast Accounting--Mandatory Response Support feature provides a mechanism to support broadcast accounting under each server group through a Gateway GPRS Support Node (GGSN), which acts as a gateway between a General Packet Radio Service (GPRS) wireless data network and other networks such as the Internet or private networks.

Prerequisites for AAA Broadcast Accounting-Mandatory Response Support

See the Cisco GGSN Release 8.0 Configuration Guide for more information on preparing for the GGSN configuration.

Restrictions for AAA Broadcast Accounting-Mandatory Response Support

Accounting information can be sent simultaneously to a maximum of ten AAA servers.

Information About AAA Broadcast Accounting-Mandatory Response Support

The AAA Broadcast Accounting--Mandatory Response Support feature allows up to 10 server groups (methods) to be configured in a method list. The following sections describe the types of AAA accounting used to support GGSN:

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple authentication, authorization, and accounting (AAA) servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of servers, which can be either RADIUS or TACACS+, and each server group can define its backup servers for failover independently of other groups. Failover is a process that may occur when more than one server has been defined within a server group. Failover refers to the process by which information is sent to the first server in a server group; if the first server is unavailable, the information is sent to the next server in the server group. This process continues until the information is successfully sent to one of the servers within the server group or until the list of available servers within the server group is exhausted.

Simultaneous Broadcast and Wait Accounting

With Cisco GGSN Release 8.0 and later releases, broadcast and wait accounting can be configured to work together. The wait accounting feature is configured at the Access Point Name (APN) level, while broadcast accounting is specified at the AAA method level.

Broadcast accounting sends start, stop, and interim accounting records to all the server groups that are configured in a method list. Within a server group, the accounting records are sent to the first active server. If the active server cannot be reached, the accounting records are sent to the next server within a group.

Additionally, one or more server groups within a method list can be configured as “mandatory,” meaning that a server from that server group has to respond to the Accounting Start message. The APN-level wait accounting ensures that an accounting response has been received from all mandatory server groups before the packet data protocol (PDP) context is established.

The advantages of broadcast and wait accounting together include:

- Accounting records are sent to multiple servers, and once the entry is made, the user can start using different services.
- Records are sent to multiple AAA servers for redundancy purposes.
- A PDP context is established only when a valid Accounting Start record has been received by all essential servers, avoiding information loss.
- Broadcast records can be sent to as many as ten server groups within a method list.

When configuring broadcast and wait accounting together, note the following:

- Under the method list configuration, the **mandatory** keyword is available only if broadcast accounting is configured.
- If wait accounting is not required, broadcast accounting to all server groups is available without any mandatory groups defined.
- If you do not specify any mandatory server groups when configuring broadcast accounting, wait accounting will function as it does in Cisco GGSN Release 7.0 and earlier releases.
- Wait accounting does not apply to PPP PDP contexts.
- A PDP is successfully created only when a Accounting response is received from all the mandatory servers.
- The periodic timer starts when an Accounting Response (PDP creation) is received.

Note More than one server group can be defined as a mandatory server group in a method list.

How AAA Broadcast Accounting is Supported for GGSN

Configuring Broadcast and Wait Accounting on the GGSN

The tasks in this section describe how to configure broadcast and wait accounting on the GGSN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa accounting network** {*method-list-name* | **default**}
5. **action-type** {**start-stop** | **stop-only**| **none**}
6. **broadcast**
7. **group** *server-group* [**mandatory**]
8. **exit**
9. **gprs access-point-list** *list-name*
10. **access-point** *access-point-index*
11. **aaa-group accounting** *method-list name*
12. **gtp-response-message wait-accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter the password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router# aaa new-model`	Enables new access control commands and functions (disables the old commands).
Step 4	**aaa accounting network** {*method-list-name* | **default**} **Example:** `Router(config)# aaa accounting network net1`	Enables authentication, authorization, and accounting (AAA) accounting of requested services for billing or security purposes when you use RADIUS and enters accounting method list mode. • The *method-list-name* argument is the named accounting list, which has a maximum of 31 characters. Any characters longer than the maximum are rejected. • The **default** keywork specifies the default accounting list.
Step 5	**action-type** {**start-stop** | **stop-only**| **none**} **Example:** `Router(cfg-acct-mlist)#action-type start-stop`	Performs a type of action on accounting records. Possible values are: • **start-stop** --Sends a “start” accounting notice at the beginning of a process and a “stop” accounting notice at the end of a process. • **stop-only** --Sends a “stop” accounting notice at the end of the requested user process. • **none** --Disables accounting services on this line or interface.
Step 6	**broadcast** **Example:** `Router(cfg-acct-mlist)#broadcast`	(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.
Step 7	**group** *server-group* [**mandatory**] **Example:** `Router(cfg-acct-mlist)#group server1`	Specifies the server group. Optionally, specify the **mandatory** keyword to define this server group as mandatory. If a server group is mandatory, a server from the server group must respond to the Accounting Start message.

	Command or Action	Purpose
		Note Up to ten server groups can be defined within a method list.
Step 8	**exit**	Exits accounting method list configuration mode.
Step 9	**gprs access-point-list** *list-name* **Example:** `Router(config)# gprs access-point-list public1`	Configures an access point list that you use to define public data network (PDN) access points on the GGSN and enters global configuration mode.
Step 10	**access-point** *access-point-index* **Example:** `Router(config-ap-list)# access-point 11`	Specifies an access point number and enters access point configuration mode.
Step 11	**aaa-group accounting** *method-list name* **Example:** `Router(config-access-point)#aaa-group accounting net1`	Specifies an accounting server group.
Step 12	**gtp-response-message wait-accounting** **Example:** `Router(config-access-point)# gtp-response-message wait-accounting`	Configures APN to wait for a RADIUS accounting response before sending a Create PDP Context response to the Serving GPRS Support Node (SGSN).

Configuration Examples for AAA Broadcast Accounting-Mandatory Response Support

AAA Broadcast Accounting-Mandatory Response Support Example

The following example globally configures the GGSN to wait for an accounting response from the RADIUS server before sending a Create PDP Context response to the SGSN. The GGSN waits for a response for PDP context requests received across all access points, except access-point 1. RADIUS response message waiting has been overridden at access-point 1 by using the **no gtp response-message wait-accounting** command.

```
! Enables AAA globally
!
aaa new-model
!
! Defines AAA server group
!
aaa group server radius abc
 server 10.2.3.4 auth-port 1645 acct-port 1646
 server 10.6.7.8 auth-port 1645 acct-port 1646
```

```
!
! Configures AAA authentication and authorization
!
aaa authentication ppp abc group abc
aaa authorization network abc group abc
aaa accounting network abc
 action-type start-stop
 broadcast
 group SG1 mandatory
 group SG2
 group SG3 mandatory
!
gprs access-point-list gprs
 access-point 1
  access-mode non-transparent
  access-point-name www.pdn1.com
  aaa-group authentication abc
!
! Disables waiting for RADIUS response
! message at APN 1
!
  no gtp response-message wait-accounting
  exit
access-point 2
 access-mode non-transparent
 access-point-name www.pdn2.com
 aaa-group authentication abc
!
! Enables waiting for RADIUS response
! messages across all APNs (except APN 1)
!
gprs gtp response-message wait-accounting
!
! Configures global RADIUS server hosts
! and specifies destination ports for
! authentication and accounting requests
!
radius-server host 10.2.3.4 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.6.7.8 auth-port 1645 acct-port 1646 non-standard
radius-server key ggsntel
```

Additional References

The following sections provide references related to the AAA Broadcast Accounting--Mandatory Response Support feature.

Related Documents

Related Topic	Document Title
Preparation for the GGSN configuration	*Cisco GGSN Release 8.0 Configuration Guide*
AAA commands	*Cisco IOS Security Command Reference Guide*
AAA features	*Cisco IOS Security Configuration Guide: Securing User Services*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

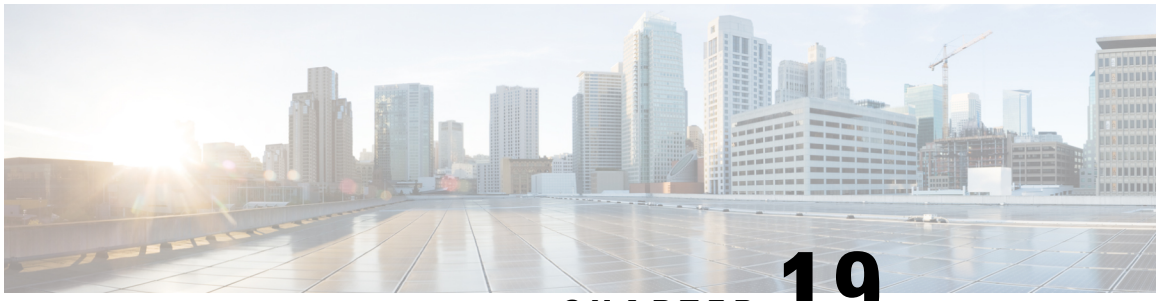
Feature Information for AAA Broadcast Accounting-Mandatory Response Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 36: Feature Information for AAA Broadcast Accounting--Mandatory Response Support

Feature Name	Releases	Feature Information
AAA Broadcast Accounting--Mandatory Response Support	Cisco IOS XE Release 3.9S	The AAA Broadcast Accounting--Mandatory Response Support feature provides a mechanism to support broadcast accounting under each server group through a Gateway GPRS Support Node (GGSN), which acts as a gateway between a General Packet Radio Service (GPRS) wireless data network and other networks such as the Internet or private networks. The following commands were introduced or modified: **aaa accounting network**, **aaa-group accounting**, **access-point**, **action-type**, **broadcast**, **gprs access-point-list**, **group**, **gtp-response-message wait-accounting**



CHAPTER 19

Password Strength and Management for Common Criteria

The Password Strength and Management for Common Criteria feature is used to specify password policies and security mechanisms for storing, retrieving, and providing rules to specify user passwords.

For local users, the user profile and the password information with the key parameters are stored on the Cisco device, and this profile is used for local authentication of users. The user can be an administrator (terminal access) or a network user (for example, PPP users being authenticated for network access).

For remote users, where the user profile information is stored in a remote server, a third-party authentication, authorization, and accounting (AAA) server may be used for providing AAA services, both for administrative and network access.

Restrictions for Password Strength and Management for Common Criteria

- Only four concurrent users can log on to the system by using vty at any moment.

Information About Password Strength and Management for Common Criteria

Password Composition Policy

The password composition policy allows you to create passwords of any combination of upper and lowercase characters, numbers, and special characters that include “!”, “@”, “#”, “$”, “%”,“^”, “&”, “*”, “(“, and “)”.

Password Length Policy

The administrator has the flexibility to set the password's minimum and maximum length. The recommended minimum password length is 8 characters. The administrator can specify both the minimum (1) and the maximum (64) length for the password.

Password Lifetime Policy

The security administrator can provide a configurable option for a password to have a maximum lifetime. If the lifetime parameter is not configured, the configured password will never expire. The maximum lifetime can be configured by providing the configurable value in years, months, days, hours, minutes, and seconds. The lifetime configuration will survive across reloads as it is a part of the configuration, but every time the system reboots, the password creation time will be updated to the new time. For example, if a password is configured with a lifetime of one month and on the 29th day, the system reboots, then the password will be valid for one month after the system reboots.

When you configure the lifetime using months, the policy sets the lifetime to 30 days regardless of the number of days in the specified month.

Password Expiry Policy

If the user attempts to log on and if the user's password credentials have expired, then the following happens:

1. The user is prompted to set the new password after successfully entering the expired password.
2. When the user enters the new password, the password is validated against the password security policy.
3. If the new password matches the password security policy, then the AAA database is updated, and the user is authenticated with the new password.
4. If the new password does not match the password security policy, then the user is prompted again for the password. From AAA perspective, there is no restriction on the number of retries. The number of retries for password prompt in case of unsuccessful authentication is controlled by the respective terminal access interactive module. For example, for telnet, after three unsuccessful attempts, the session will be terminated.

If the password's lifetime is not configured for a user and the user has already logged on and if the security administrator configures the lifetime for that user, then the lifetime will be set in the database. When the same user is authenticated the next time, the system will check for password expiry. The password expiry is checked only during the authentication phase.

If the user has been already authenticated and logged on to the system and if the password expires, then no action will be taken. The user will be prompted to change the password only during the next authentication for the same user.

Password Change Policy

The new password must contain a minimum of 4 character changes from the previous password. A password change can be triggered by the following scenarios:

- The security administrator wants to change the password.
- The user is trying to get authenticated using a profile, and the password for that profile has expired.

When the security administrator changes the password security policy and the existing profile does not meet the password security policy rules, no action will be taken if the user has already logged on to the system. The user will be prompted to change the password only when the user tries to get authenticated using the profile that does not meet the password security restriction.

When the user changes the password, the lifetime parameters set by the security administrator for the old profile will be the lifetime parameters for the new password.

For noninteractive clients such as dot1x, when the password expires, appropriate error messages will be sent to the clients, and the clients must contact the security administrator to renew the password.

User Reauthentication Policy

Users are reauthenticated when they change their passwords.

When users change their passwords on expiry, they will be authenticated against the new password. In such cases, the actual authentication happens based on the previous credentials, and the new password is updated in the database.

Note Users can change their passwords only when they are logging on and after the expiry of the old password; however, a security administrator can change the user's password at any time.

Support for Framed (noninteractive) Session

When a client such as dot1x uses the local database for authentication, the Password Strength and Management for Common Criteria feature will be applicable; however, upon password expiry, clients will not be able to change the password. An appropriate failure message will be sent to such clients, and the user must request the security administrator to change the password.

How to Configure Password Strength and Management for Common Criteria

Configuring the Password Security Policy

Perform this task to create a password security policy and to apply the policy to a specific user profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa common-criteria policy** *policy-name*
5. **char-changes** *number*
6. **max-length** *number*
7. **min-length** *number*
8. **numeric-count** *number*
9. **special-case** *number*
10. **exit**
11. **username** *username* **common-criteria-policy** *policy-name* **password** *password*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa common-criteria policy** *policy-name* **Example:**	Creates the AAA security password policy and enters common criteria configuration policy mode.

	Command or Action	Purpose
	`Device(config)# aaa common-criteria policy policy1`	
Step 5	**char-changes** *number* **Example:** `Device(config-cc-policy)# char-changes 4`	(Optional) Specifies the number of changed characters between old and new passwords.
Step 6	**max-length** *number* **Example:** `Device(config-cc-policy)# max-length 25`	(Optional) Specifies the maximum length of the password.
Step 7	**min-length** *number* **Example:** `Device(config-cc-policy)# min-length 8`	(Optional) Specifies the minimum length of the password.
Step 8	**numeric-count** *number* **Example:** `Device(config-cc-policy)# numeric-count 4`	(Optional) Specifies the number of numeric characters in the password.
Step 9	**special-case** *number* **Example:** `Device(config-cc-policy)# special-case 3`	(Optional) Specifies the number of special characters in the password.
Step 10	**exit** **Example:** `Device(config-cc-policy)# exit`	(Optional) Exits common criteria configuration policy mode and returns to global configuration mode.
Step 11	**username** *username* **common-criteria-policy** *policy-name* **password** *password* **Example:** `Device(config)# username user1 common-criteria-policy policy1 password password1`	(Optional) Applies a specific policy and password to a user profile.
Step 12	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying the Common Criteria Policy

Perform this task to verify all the common criteria security policies.

SUMMARY STEPS

1. **enable**
2. **show aaa common-criteria policy name** *policy-name*
3. **show aaa common-criteria policy all**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show aaa common-criteria policy name *policy-name*

Displays the password security policy information for a specific policy.

Example:

```
Device# show aaa common-criteria policy name policy1

Policy name: policy1
Minimum length: 1
Maximum length: 64
Upper Count: 20
Lower Count: 20
Numeric Count: 5
Special Count: 2
Number of character changes 4
Valid forever. User tied to this policy will not expire.
```

Step 3 show aaa common-criteria policy all

Displays password security policy information for all the configured policies.

Example:

```
Device# show aaa common-criteria policy all
==========================================================================
Policy name: policy1
Minimum length: 1
Maximum length: 64
Upper Count: 20
Lower Count: 20
Numeric Count: 5
Special Count: 2
Number of character changes 4
Valid forever. User tied to this policy will not expire.
==========================================================================
Policy name: policy2
Minimum length: 1
Maximum length: 34
Upper Count: 10
```

```
Lower Count: 5
Numeric Count: 4
Special Count: 2
Number of character changes 2
Valid forever. User tied to this policy will not expire.
=========================================================================
```

Troubleshooting Tips

Use the **debug aaa common-criteria** command to troubleshoot AAA common criteria.

Configuration Example for the Password Strength and Management for Common Criteria Feature

Example: Password Strength and Management for Common Criteria

The following example shows how to create a common criteria security policy and apply the specific policy to a user profile:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa common-criteria policy policy1
Device(config-cc-policy)# char-changes 4
Device(config-cc-policy)# max-length 20
Device(config-cc-policy)# min-length 6
Device(config-cc-policy)# numeric-count 2
Device(config-cc-policy)# special-case 2
Device(config-cc-policy)# exit
Device(config)# username user1 common-criteria-policy policy1 password password1
Device(config)# end
```

Additional References

The following sections provide references related to the RADIUS Packet of Disconnect feature.

Related Documents

Related Topic	Document Title
AAA	Authentication, Authorization, and Accounting (AAA) section of the *Cisco IOS XE Security Configuration Guide, Securing User Services*, Release 2.
Security commands	*Cisco IOS Security Command Reference*
CLI Configuration	*Cisco IOS XE Configuration Fundamentals Configuration Guide*, Release 2

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial-in User Service*
RFC 3576	*Dynamic Authorization Extensions to RADIUS*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

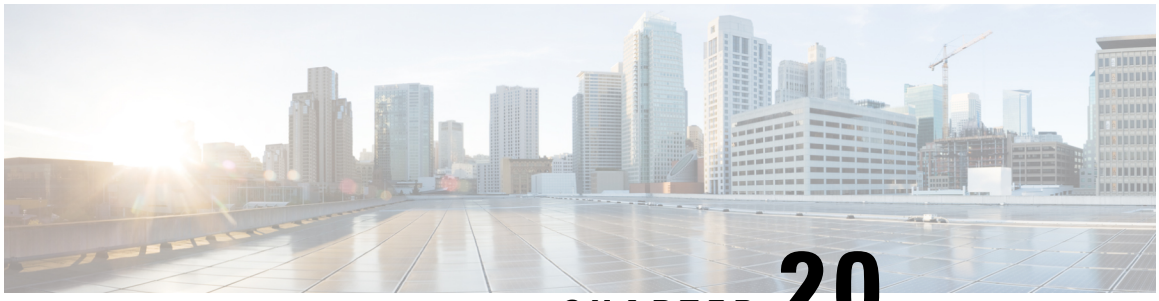
Feature Information for Password Strength and Management for Common Criteria

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 37: Feature Information for Password Strength and Management for Common Criteria

Feature Name	Releases	Feature Information
Password Strength and Management for Common Criteria		The Password Strength and Management for Common Criteria feature is used to specify password policies and security mechanisms for storing, retrieving, and providing rules to specify user passwords. The following commands were introduced or modified: **aaa common-criteria policy**, **debug aaa common-criteria**, and **show aaa common-criteria policy**.



CHAPTER 20

Secure Reversible Passwords for AAA

The Secure Reversible Passwords for AAA feature enables secure reversible encryption for authentication, authorization, and accounting (AAA) configurations using type 6 advanced encryption scheme (AES) passwords.

Prerequisites for Secure Reversible Passwords for AAA

The following commands should be enabled for the type 6 password encryption:

- **password encryption aes**
- **key config-key password-encrypt** [*password*]
- **aaa new-model**

Information About Secure Reversible Passwords for AAA

Secure Reversible Passwords

Passwords in Cisco IOS configurations require a secure storage so that the key for the reversible encryption can be stored to ensure that authentication methods can access the user credentials whenever required.

Reversible encryption is the process by which a password is encrypted with a reversible, symmetric encryption algorithm. To check if the password entered by the user is valid, the password is decrypted and compared to the user-input password. To perform this encryption, the symmetric encryption algorithm requires a key.

The type 6 advanced encryption scheme (AES) encrypted passwords help to secure the reversible passwords for authentication, authorization, and accounting (AAA) features. This type 6 encryption key is stored in a private NVRAM and secured.

AAA network configurations use Lightweight Directory Access Protocol (LDAP), RADIUS, or TACACS+ server hosts. Use the **radius server host**, **tacacs-server host**, and **ldap server** commands to configure RADIUS, TACACS+, or LDAP host servers respectively.

Type 6 Encryption Configuration

The following commands have been updated with the type **6** keyword to enable secure reversible passwords to configure authentication, authorization, and accounting (AAA) features. For more information about the security commands, see the *Cisco IOS Security Command Reference*. For more information about AAA configuration, see the *Authentication, Authorization, and Accounting Configuration Guide*.

- **aaa configuration**
  - **aaa configuration** {**config-username username** *username* [**password** [**0** | **7**] *password*] | {**pool** | **route**} **username** *username* [**password** [**0** | **6** | **7**] *password*}
- **bind authenticate root-dn (config-ldap-server)**
  - **bind authenticate root-dn** *username* **password** {**0** *string* | **6** *string* | **7** *string*} *string*
- **client (config-locsvr-da-radius)**
  - **client** *ip-address* **server-key** [**0** | **6** | **7**] *word*
- **key (config-radius-server)**
  - **key** {**0** *string* | **6** *string* | **7** *string*} *string*
- **key (config-server-tacacs)**
  - **key** {**0** *string* | **6** *string* | **7** *string*} *string*
- **pac key (config-radius-server)**
  - **pac key** {**0** *string* | **6** *string* | **7** *string*} *string*
- **password (config-filter)**
  - **password** [**0** | **6** | **7**] *password*
- **server-private (RADIUS)**
  - **server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** [**0** | **6** | **7**] *string*]
- **server-private (TACACS+)**
  - **server-private** {*ip-address* | *name* | *ipv6-address*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **6** | **7**] *string*]
- **tacacs-server host**
  - **tacacs-server host** {*host-name* | *host-ip-address*} [**key** {**0** *string* | **6** *string* | **7** *string*} *string*] [[**nat**] [**port** [*integer*]] [**single-connection**] [**timeout** [*integer*]]]

- **tacacas-server key**
  - **tacacs-server key** {**0** *string* | **6** *string* | **7** *string*} *string*

Additional References for Secure Reversible Passwords for AAA

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
AAA configuration	*Authentication, Authorization, and Accounting Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Secure Reversible Passwords for AAA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

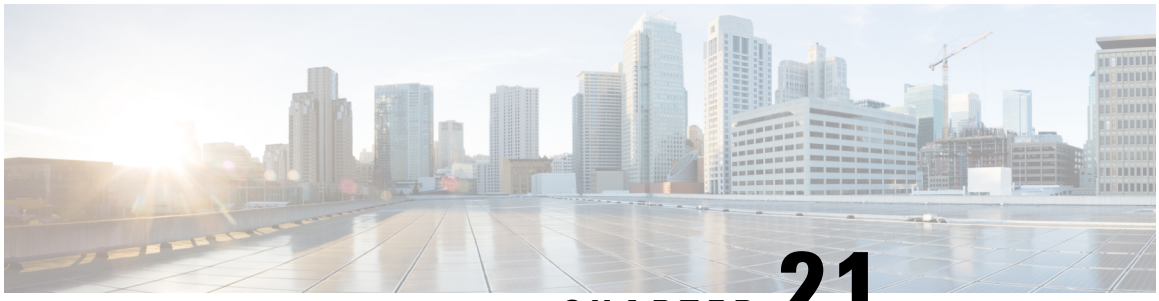
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for Secure Reversible Passwords for AAA

Feature Name	Releases	Feature Information
Secure Reversible Passwords for AAA	15.4(1)T	The Secure Reversible Passwords for AAA feature enables secure reversible encryption for authentication, authorization, and accounting (AAA) configurations using type 6 advanced encryption scheme (AES) passwords. The following commands were introduced or modified: **aaa configuration**, **bind authenticate root-dn (config-ldap-server)**, **client (config-locsvr-da-radius)**, **key (config-radius-server)**, **key (config-server-tacacs)**, **pac key (config-radius-server)**, **password (config-filter)**, **server-private (RADIUS)**, **server-private (TACACS+)**, **tacacs-server host**, and **tacacas-server key**.

PART II

Secure Shell



CHAPTER 21

Reverse SSH Enhancements

The Reverse SSH Enhancements feature, which is supported for SSH Version 1 and 2, provides an alternative way to configure reverse Secure Shell (SSH) so that separate lines do not need to be configured for every terminal or auxiliary line on which SSH must be enabled. This feature also eliminates the rotary-group limitation.

Prerequisites for Reverse SSH Enhancements

- SSH must be enabled.
- The SSH client and server must be running the same version of SSH.

Restrictions for Reverse SSH Enhancements

- The **-l** keyword and *userid* **:**{*number*} {*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

Information About Reverse SSH Enhancements

Reverse Telnet

Reverse telnet allows you to telnet to a certain port range and connect to terminal or auxiliary lines. Reverse telnet has often been used to connect a Cisco device that has many terminal lines to the consoles of other Cisco devices. Telnet makes it easy to reach the device console from anywhere simply by telnet to the terminal

server on a specific line. This telnet approach can be used to configure a device even if all network connectivity to that device is disconnected. Reverse telnet also allows modems that are attached to Cisco devices to be used for dial-out (usually with a rotary device).

Reverse SSH

Reverse telnet can be accomplished using SSH. Unlike reverse telnet, SSH provides for secure connections. The Reverse SSH Enhancements feature provides you with a simplified method of configuring SSH. Using this feature, you no longer have to configure a separate line for every terminal or auxiliary line on which you want to enable SSH. The previous method of configuring reverse SSH limited the number of ports that can be accessed to 100. The Reverse SSH Enhancements feature removes the port number limitation. For information on the alternative method of configuring reverse SSH, see How to Configure Reverse SSH Enhancements, on page 260.

How to Configure Reverse SSH Enhancements

Configuring Reverse SSH for Console Access

To configure reverse SSH console access on the SSH server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number ending-line-number*
4. **no exec**
5. **login authentication** *listname*
6. **transport input ssh**
7. **exit**
8. **exit**
9. **ssh -l** *userid* **:** {*number*} {*ip-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**line** *line-number ending-line-number* **Example:** `Device# line 1 3`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Device(config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Device(config-line)# login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.
Step 6	**transport input ssh** **Example:** `Device(config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the device. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 7	**exit** **Example:** `Device(config-line)# exit`	Exits line configuration mode.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 9	**ssh -l** *userid* **:** {*number*} {*ip-address*} **Example:** `Device# ssh -l lab:1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid* --User ID. • **:** --Signifies that a port number and terminal IP address will follow the userid argument. • *number* --Terminal or auxiliary line number. • *ip-address* --Terminal server IP address. **Note** The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Configuring Reverse SSH for Modem Access

To configure Reverse SSH for modem access, perform the steps shown in the "SUMMARY STEPS" section below.

In this configuration, reverse SSH is being configured on a modem used for dial-out lines. To get any of the dial-out modems, you can use any SSH client and start a SSH session as shown (in Step 10) to get to the next available modem from the rotary device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number ending-line-number*
4. **no exec**
5. **login authentication** *listname*
6. **rotary** *group*
7. **transport input ssh**
8. **exit**
9. **exit**
10. **ssh -l** *userid* **:rotary** {*number*} {*ip-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line** *line-number ending-line-number* **Example:** `Device# line 1 200`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Device(config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Device(config-line)# login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.

	Command or Action	Purpose
Step 6	**rotary** *group* **Example:** `Device(config-line)# rotary 1`	Defines a group of lines consisting of one or more virtual terminal lines or one auxiliary port line.
Step 7	**transport input ssh** **Example:** `Device(config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the device. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 8	**exit** **Example:** `Device(config-line)# exit`	Exits line configuration mode.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 10	**ssh -l** *userid* **:rotary** {*number*} {*ip-address*} **Example:** `Device# ssh -l lab:rotary1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid* --User ID. • **:** --Signifies that a port number and terminal IP address will follow the *userid* argument. • *number* --Terminal or auxiliary line number. • *ip-address* --Terminal server IP address. **Note** The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Troubleshooting Reverse SSH on the Client

To troubleshoot the reverse SSH configuration on the client (remote device), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh client**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh client** **Example:** `Device# debug ip ssh client`	Displays debugging messages for the SSH client.

Troubleshooting Reverse SSH on the Server

To troubleshoot the reverse SSH configuration on the terminal server, perform the following steps. The steps may be configured in any order or independent of one another.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**
3. **show ssh**
4. **show line**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh** **Example:** `Device# debug ip ssh`	Displays debugging messages for the SSH server.
Step 3	**show ssh** **Example:** `Device# show ssh`	Displays the status of the SSH server connections.
Step 4	**show line** **Example:** `Device# show line`	Displays parameters of a terminal line.

Configuration Examples for Reverse SSH Enhancements

Example Reverse SSH Console Access

The following configuration example shows that reverse SSH has been configured for console access for terminal lines 1 through 3:

Terminal Server Configuration

```
line 1 3
   no exec
   login authentication default
   transport input ssh
```

Client Configuration

The following commands configured on the SSH client will form the reverse SSH session with lines 1, 2, and 3, respectively:

```
ssh -l lab:1 router.example.com
ssh -l lab:2 router.example.com
ssh -l lab:3 router.example.com
```

Example Reverse SSH Modem Access

The following configuration example shows that dial-out lines 1 through 200 have been grouped under rotary group 1 for modem access:

```
line 1 200
   no exec
   login authentication default
   rotary 1
   transport input ssh
   exit
```

The following command shows that reverse SSH will connect to the first free line in the rotary group:

```
ssh -l lab:rotary1 router.example.com
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring Secure Shell	Secure Shell Configuration Guide

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring Secure Shell	Secure Shell Configuration Guide
Security commands	Cisco IOS Security Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature.	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

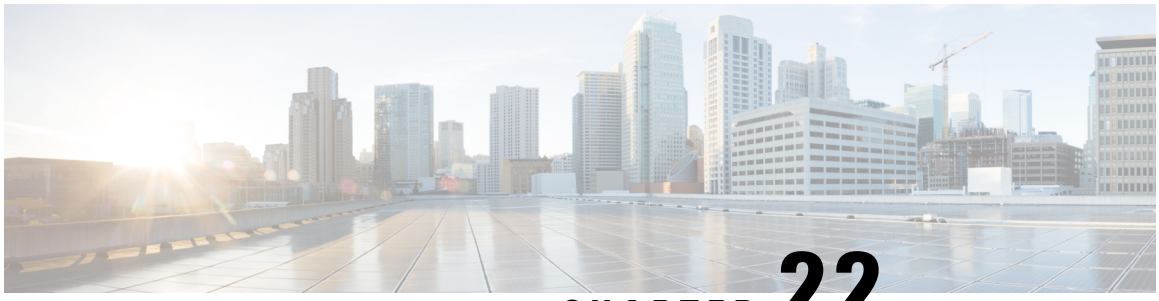
Feature Information for Reverse SSH Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 39: Feature Information for Reverse SSH Enhancements

Feature Name	Releases	Feature Information
Reverse SSH Enhancements		The Reverse SSH Enhancements feature, which is supported for SSH Version 1 and 2, provides an alternative way to configure reverse Secure Shell (SSH) so that separate lines do not need to be configured for every terminal or auxiliary line on which SSH must be enabled. This feature also eliminates the rotary-group limitation. The following command was introduced: **ssh**.



CHAPTER 22

Secure Copy

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools.

Prerequisites for Secure Copy

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.
- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

Restrictions for Secure Copy Performance Improvement

- Incrementing window-size must be used mainly for SCP operations only.
- Depending on the platform type, the maximum window size can cause high CPU usage.
- As a precaution, increments can be made up to four times the default size.

Information About Secure Copy

How SCP Works

The behavior of SCP is similar to that of remote copy (rcp), which comes from the Berkeley r-tools suite, except that SCP relies on SSH for security. In addition, SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.

SCP allows a user who has appropriate authorization to copy any file that exists in the Cisco IOS XE File System (IFS) to and from a router by using the **copy** command. An authorized administrator may also perform this action from a workstation.

How to Configure SCP

Configuring SCP

To enable and configure a Cisco router for SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1*[*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
6. **username** *name* [**privilege** *level]*{**password** *encryption-type encrypted-password*}
7. **ip scp server enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1*[*method2...*] **Example:** `Router (config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Router (config)# aaa authorization exec default group tacacs+`	Sets parameters that restrict user access to a network. **Note** **The exec** keyword runs authorization to determine if the user is allowed to run an EXEC shell; therefore, you must use it when you configure SCP.
Step 6	**username** *name* [**privilege** *level]*{**password** *encryption-type encrypted-password*} **Example:** `Router (config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You may skip this step if a network-based authentication mechanism--such as TACACS+ or RADIUS--has been configured.
Step 7	**ip scp server enable** **Example:** `Router (config)# ip scp server enable`	Enables SCP server-side functionality.

Verifying SCP

To verify SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**show running-config** **Example:** `Router# show running-config`	Verifies the SCP server-side functionality.

Troubleshooting SCP

SUMMARY STEPS

1. **enable**
2. **debug ip scp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip scp** **Example:** `Router# debug ip scp`	Troubleshoots SCP authentication problems.

Configuration Examples for Secure Copy

Example SCP Server-Side Configuration Using Local Authentication

The following example shows how to configure the server-side functionality of SCP. This example uses a locally defined username and password.

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username tiger privilege 15 password 0 lab
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Example SCP Server-Side Configuration Using Network-Based Authentication

The following example shows how to configure the server-side functionality of SCP using a network-based authentication mechanism:

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Secure Shell	Configuring Secure Shell and Secure Shell Version 2 Support feature modules.
Configuring authentication and authorization	Configuring Authentication , Configuring Authorization , and Configuring Accounting feature modules.

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Secure Copy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 40: Feature Information for Secure Copy

Feature Name	Releases	Feature Configuration Information
Secure Copy	Cisco IOS XE Release 2.1	The Secure Copy (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **debug ip scp**, **ip scp server enable**.

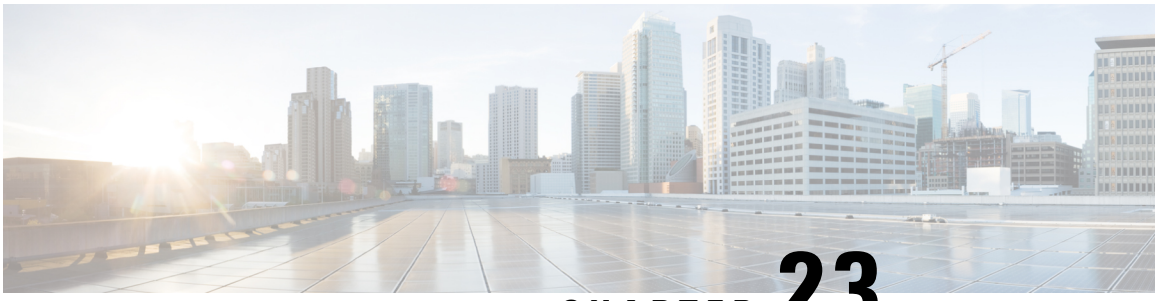
Glossary

AAA --authentication, authorization, and accounting. Framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

rcp --remote copy. Relying on Remote Shell (Berkeley r-tools suite) for security, rcp copies files, such as router images and startup configurations, to and from routers.

SCP --secure copy. Relying on SSH for security, SCP support allows the secure and authenticated copying of anything that exists in the Cisco IOS XE File Systems. SCP is derived from rcp.

SSH --Secure Shell. Application and a protocol that provide a secure replacement for the Berkeley r-tools. The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. SSH Version 1 is implemented in the Cisco IOS XE software.



CHAPTER 23

Secure Shell Version 2 Support

The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2. (SSH Version 1 support was implemented in an earlier Cisco software release.) SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. The only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer over a network. The Secure Copy Protocol (SCP) feature that is provided with SSH allows for the secure transfer of files.

Prerequisites for Secure Shell Version 2 Support

- Before configuring SSH, ensure that the required image is loaded on your device. The SSH server requires you to have a k9 (Triple Data Encryption Standard [3DES]) software image depending on your release.
- You have to use a SSH remote device that supports SSH Version 2 and connect to a Cisco device.
- SCP relies on authentication, authorization, and accounting (AAA) to function correctly. Therefore, AAA must be configured on the device to enable the secure copy protocol on the SSH Server.

Note The SSH Version 2 server and the SSH Version 2 client are supported on your Cisco software, depending on your release. (The SSH client runs both the SSH Version 1 protocol and the SSH Version 2 protocol. The SSH client is supported in both k8 and k9 images depending on your release.)

For more information about downloading a software image, refer to the *Configuration Fundamentals Configuration Guide*.

Restrictions for Secure Shell Version 2 Support

- From Cisco IOS XE Release 17.10, the Secure Shell Version 1.99 is not supported.
- Secure Shell (SSH) servers and SSH clients are supported in Triple Data Encryption Standard (3DES) software images.
- Execution Shell, remote command execution, and Secure Copy Protocol (SCP) are the only applications supported.
- Rivest, Shamir, and Adleman (RSA) key generation is an SSH server-side requirement. Devices that act as SSH clients need not generate RSA keys.
- From Cisco IOS XE Release 17.10, the minimum RSA key pair size must be 2048 bits.

  From Cisco IOS XE Release 17.11, if you want to continue using the weak RSA key, disable CSDL compliance on the device using the **crypto engine compliance shield disable** command, and reboot.
- The following features are not supported:
  - Port forwarding
  - Compression

Information About Secure Shell Version 2 Support

Secure Shell Version 2

The Secure Shell Version 2 Support feature allows you to configure SSH Version 2.

The configuration for the SSH Version 2 server is similar to the configuration for SSH Version 1. The **ip ssh version** command defines the SSH version to be configured. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH Version 1 and SSH Version 2 connections are honored.

Note SSH Version 1 is a protocol that has never been defined in a standard. If you do not want your device to fall back to the undefined protocol (Version 1), you should use the **ip ssh version** command and specify Version 2.

The **ip ssh rsa keypair-name** command enables an SSH connection using the Rivest, Shamir, and Adleman (RSA) keys that you have configured. Previously, SSH was linked to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). This behavior still exists, but by using the **ip ssh rsa keypair-name** command, you can overcome this behavior. If you configure the **ip ssh rsa keypair-name** command with a key pair name, SSH is enabled if the key pair exists or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you are not forced to configure a hostname and a domain name, which was required in SSH Version 1 of the Cisco software.

Note The login banner is supported in SSH Version 2, but it is not supported in Secure Shell Version 1.

Secure Shell Version 2 Enhancements

The SSH Version 2 Enhancements feature includes a number of additional capabilities such as supporting Virtual Routing and Forwarding (VRF)-Aware SSH, SSH debug enhancements, and Diffie-Hellman (DH) group exchange support.

Note The VRF-Aware SSH feature is supported depending on your release.

The Cisco SSH implementation has traditionally used 768-bit modulus, but with an increasing need for higher key sizes to accommodate DH Group 14 (2048 bits) and Group 16 (4096 bits) cryptographic applications, a message exchange between the client and the server to establish the favored DH group becomes necessary. The **ip ssh dh min size** command configures the modulus size on the SSH server. In addition to this, the **ssh** command was extended to add VRF awareness to the SSH client-side functionality through which the VRF instance name in the client is provided with the IP address to look up the correct routing table and establish a connection.

Debugging was enhanced by modifying SSH debug commands. The **debug ip ssh** command was extended to simplify the debugging process. Before the simplification of the debugging process, this command printed all debug messages related to SSH regardless of what was specifically required. The behavior still exists, but if you configure the **debug ip ssh** command with a keyword, messages are limited to information specified by the keyword.

Secure Shell Version 2 Enhancements for RSA Keys

Cisco SSH Version 2 supports keyboard-interactive and password-based authentication methods. The SSH Version 2 Enhancements for RSA Keys feature also supports RSA-based public key authentication for the client and the server.

User authentication—RSA-based user authentication uses a private/public key pair associated with each user for authentication. The user must generate a private/public key pair on the client and configure a public key on the Cisco SSH server to complete the authentication.

An SSH user trying to establish credentials provides an encrypted signature using the private key. The signature and the user's public key are sent to the SSH server for authentication. The SSH server computes a hash over the public key provided by the user. The hash is used to determine if the server has a matching entry. If a match is found, an RSA-based message verification is performed using the public key. Hence, the user is authenticated or denied access based on the encrypted signature.

Server authentication—While establishing an SSH session, the Cisco SSH client authenticates the SSH server by using the server host keys available during the key exchange phase. SSH server keys are used to identify the SSH server. These keys are created at the time of enabling SSH and must be configured on the client.

For server authentication, the Cisco SSH client must assign a host key for each server. When the client tries to establish an SSH session with a server, the client receives the signature of the server as part of the key exchange message. If the strict host key checking flag is enabled on the client, the client checks if it has the host key entry corresponding to the server. If a match is found, the client tries to validate the signature by

using the server host key. If the server is successfully authenticated, the session establishment continues; otherwise, it is terminated and displays a "Server Authentication Failed" message.

Note Storing public keys on a server uses memory; therefore, the number of public keys configurable on an SSH server is restricted to ten users, with a maximum of two public keys per user.

Note RSA-based user authentication is supported by the Cisco server, but Cisco clients cannot propose public key as an authentication method. If the Cisco server receives a request from an open SSH client for RSA-based authentication, the server accepts the authentication request.

Note For server authentication, configure the RSA public key of the server manually and configure the **ip ssh stricthostkeycheck** command on the Cisco SSH client.

SNMP Trap Generation

Depending on your release, Simple Network Management Protocol (SNMP) traps are generated automatically when an SSH session terminates if the traps have been enabled and SNMP debugging has been enabled. For information about enabling SNMP traps, see the "Configuring SNMP Support" module in the *SNMP Configuration Guide*.

Note When you configure the **snmp-server host** command, the IP address must be the address of the PC that has the SSH (telnet) client and that has IP connectivity to the SSH server.

You must also enable SNMP debugging using the **debug snmp packet** command to display the traps. The trap information includes information such as the number of bytes sent and the protocol that was used for the SSH session.

SSH Keyboard Interactive Authentication

The SSH Keyboard Interactive Authentication feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature. The feature is automatically enabled.

The following methods are supported:

- Password
- SecurID and hardware tokens printing a number or a string in response to a challenge sent by the server
- Pluggable Authentication Module (PAM)
- S/KEY (and other One-Time-Pads)

For examples of various scenarios in which the SSH Keyboard Interactive Authentication feature has been automatically enabled, see the Examples: SSH Keyboard Interactive Authentication, on page 296 section.

How to Configure Secure Shell Version 2 Support

Configuring a Device for SSH Version 2 Using a Hostname and Domain Name

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version** [**1** | **2**]
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *name* **Example:** `Device(config)# hostname cisco7200`	Configures a hostname for your device.
Step 4	**ip domain-name** *name* **Example:** `cisco7200(config)# ip domain-name example.com`	Configures a domain name for your device.
Step 5	**crypto key generate rsa** **Example:** `cisco7200(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.

	Command or Action	Purpose
Step 6	**ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*] **Example:** `cisco7200(config)# ip ssh time-out 120`	(Optional) Configures SSH control variables on your device.
Step 7	**ip ssh version** [**1** | **2**] **Example:** `cisco7200(config)# ip ssh version 1`	(Optional) Specifies the version of SSH to be run on your device.
Step 8	**exit** **Example:** `cisco7200(config)# exit`	Exits global configuration mode and enters privileged EXEC mode. • Use **no hostname** command to return to the default host.

Configuring a Device for SSH Version 2 Using RSA Key Pairs

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip ssh rsa keypair-name** *keypair-name*

Example:

```
Device(config)# ip ssh rsa keypair-name sshkeys
```

Specifies the RSA key pair to be used for SSH.

Note A Cisco device can have many RSA key pairs.

Step 4 **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*

Example:

```
Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768
```

Enables the SSH server for local and remote authentication on the device.

- For SSH Version 2, the modulus size must be at least 768 bits.

Note To delete the RSA key pair, use the **crypto key zeroize rsa** command. When you delete the RSA key pair, you automatically disable the SSH server.

Step 5 **ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*]

Example:

```
Device(config)# ip ssh time-out 12
```

Configures SSH control variables on your device.

Step 6 **ip ssh version 2**

Example:

```
Device(config)# ip ssh version 2
```

Specifies the version of SSH to be run on the device.

Step 7 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and enters privileged EXEC mode.

Configuring the Cisco SSH Server to Perform RSA-Based User Authentication

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **hostname** *name*

Example:

```
Device(config)# hostname host1
```

Specifies the hostname.

Step 4 **ip domain-name** *name*

Example:

```
host1(config)# ip domain-name name1
```

Defines a default domain name that the Cisco software uses to complete unqualified hostnames.

Step 5 **crypto key generate rsa**

Example:

```
host1(config)# crypto key generate rsa
```

Generates RSA key pairs.

Step 6 **ip ssh pubkey-chain**

Example:

```
host1(config)# ip ssh pubkey-chain
```

Configures SSH-RSA keys for user and server authentication on the SSH server and enters public-key configuration mode.

- The user authentication is successful if the RSA public key stored on the server is verified with the public or the private key pair stored on the client.

Step 7 **username** *username*

Example:

```
host1(conf-ssh-pubkey)# username user1
```

Configures the SSH username and enters public-key user configuration mode.

Step 8 **key-string**

Example:

```
host1(conf-ssh-pubkey-user)# key-string
```

Specifies the RSA public key of the remote peer and enters public-key data configuration mode.

Note You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file.

Step 9 **key-hash** *key-type key-name*

Example:

```
host1(conf-ssh-pubkey-data)# key-hash ssh-rsa key1
```

(Optional) Specifies the SSH key type and version.

- The key type must be ssh-rsa for the configuration of private public key pairs.

- This step is optional only if the **key-string** command is configured.
- You must configure either the **key-string** command or the **key-hash** command.

Note You can use a hashing software to compute the hash of the public key string, or you can also copy the hash value from another Cisco device. Entering the public key data using the **key-string** command is the preferred way to enter the public key data for the first time.

Step 10 **end**

Example:

```
host1(conf-ssh-pubkey-data)# end
```

Exits public-key data configuration mode and returns to privileged EXEC mode.

- Use **no hostname** command to return to the default host.

Configuring the Cisco IOS SSH Client to Perform RSA-Based Server Authentication

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **hostname** *name*

Example:

```
Device(config)# hostname host1
```

Specifies the hostname.

Step 4 **ip domain-name** *name*

Example:

```
host1(config)# ip domain-name name1
```

Defines a default domain name that the Cisco software uses to complete unqualified hostnames.

Step 5 **crypto key generate rsa**

Example:

```
host1(config)# crypto key generate rsa
```

Generates RSA key pairs.

Step 6 **ip ssh pubkey-chain**

Example:

```
host1(config)# ip ssh pubkey-chain
```

Configures SSH-RSA keys for user and server authentication on the SSH server and enters public-key configuration mode.

Step 7 **server** *server-name*

Example:

```
host1(conf-ssh-pubkey)# server server1
```

Enables the SSH server for public-key authentication on the device and enters public-key server configuration mode.

Step 8 **key-string**

Example:

```
host1(conf-ssh-pubkey-server)# key-string
```

Specifies the RSA public-key of the remote peer and enters public key data configuration mode.

Note You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file.

Step 9 **exit**

Example:

```
host1(conf-ssh-pubkey-data)# exit
```

Exits public-key data configuration mode and enters public-key server configuration mode.

Step 10 **key-hash** *key-type key-name*

Example:

```
host1(conf-ssh-pubkey-server)# key-hash ssh-rsa key1
```

(Optional) Specifies the SSH key type and version.

- The key type must be ssh-rsa for the configuration of private/public key pairs.
- This step is optional only if the **key-string** command is configured.
- You must configure either the **key-string** command or the **key-hash** command.

Note You can use a hashing software to compute the hash of the public key string, or you can copy the hash value from another Cisco device. Entering the public key data using the **key-string** command is the preferred way to enter the public key data for the first time.

Step 11 **end**

Example:

```
host1(conf-ssh-pubkey-server)# end
```

Exits public-key server configuration mode and returns to privileged EXEC mode.

Step 12 **configure terminal**

Example:

```
host1# configure terminal
```

Enters global configuration mode.

Step 13 **ip ssh stricthostkeycheck**

Example:

```
host1(config)# ip ssh stricthostkeycheck
```

Ensures that server authentication takes place.

- The connection is terminated in case of a failure.
- Use **no hostname** command to return to the default host.

Starting an Encrypted Session with a Remote Device

Note The device with which you want to connect must support a Secure Shell (SSH) server that has an encryption algorithm that is supported in Cisco software. Also, you need not enable your device. SSH can be run in disabled mode.

ssh [**-v** {**1** | **2**} | **-c** {**aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-cbc** | **3des** | **aes192-cbc** | **aes256-cbc**} | **-l** *user-id* | **-l** *user-id***:***vrf-name number ip-address ip-address* | **-l** *user-id***:***rotary number ip-address* | **-m** {**hmac-md5-128** | **hmac-md5-96** | **hmac-sha1-160** | **hmac-sha1-96**} | **-o numberofpasswordprompts** *n* | **-p** *port-num*] {*ip-addr* | *hostname*} [**command** | **-vrf**]

Example:

```
Device# ssh -v 2 -c aes256-ctr -m hmac-sha1-96 -l user2 10.76.82.24
```

Starts an encrypted session with a remote networking device.

Troubleshooting Tips

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine the SSH version that has a problem.

Enabling Secure Copy Protocol on the SSH Server

Note The following task configures the server-side functionality for SCP. This task shows a typical configuration that allows the device to securely copy files from a remote workstation.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **aaa new-model**

Example:

```
Device(config)# aaa new-model
```

Enables the AAA access control model.

Step 4 **aaa authentication login default local**

Example:

```
Device(config)# aaa authentication login default local
```

Sets AAA authentication at login to use the local username database for authentication.

Step 5 **aaa authorization exec defaultlocal**

Example:

```
Device(config)# aaa authorization exec default local
```

Sets the parameters that restrict user access to a network, runs the authorization to determine if the user ID is allowed to run an EXEC shell, and specifies that the system must use the local database for authorization.

Step 6 **username***name* **privilege** *privilege-level* **password** *password*

Example:

```
Device(config)# username samplename privilege 15 password password1
```

Establishes a username-based authentication system, and specifies the username, privilege level, and an unencrypted password.

Note The minimum value for the *privilege-level* argument is 15. A privilege level of less than 15 results in the connection closing.

Step 7 **ip ssh time-out***seconds*

Example:

```
Device(config)# ip ssh time-out 120
```

Sets the time interval (in seconds) that the device waits for the SSH client to respond.

Step 8 **ip ssh authentication-retries** *integer*

Example:

```
Device(config)# ip ssh authentication-retries 3
```

Sets the number of authentication attempts after which the interface is reset.

Step 9 **ip scpserverenable**

Example:

```
Device(config)# ip scp server enable
```

Enables the device to securely copy files from a remote workstation.

Step 10 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Step 11 **debug ip scp**

Example:

```
Device# debug ip scp
```

(Optional) Provides diagnostic information about SCP authentication problems.

Verifying the Status of the Secure Shell Connection

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ssh**

Example:

```
Device# show ssh
```

Displays the status of SSH server connections.

Step 3 **exit**

Example:

```
Device# exit
```

Exits privileged EXEC mode and returns to user EXEC mode.

Examples

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for Version 1 and Version 2 connections:

```
-------------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
Connection Version Mode Encryption  Hmac               State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
-------------------------------------------------------------------------
```

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for a Version 2 connection with no Version 1 connection:

```
-------------------------------------------------------------------------
Device# show ssh

Connection Version Mode Encryption  Hmac               State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
```

```
%No SSHv1 server connections running.
-----------------------------------------------------------------------
```

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for a Version 1 connection with no Version 2 connection:

```
-----------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
%No SSHv2 server connections running.
-----------------------------------------------------------------------
```

Verifying the Secure Shell Status

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip ssh**

Example:

```
Device# show ip ssh
```

Displays the version and configuration data for SSH.

Step 3 **exit**

Example:

```
Device# exit
```

Exits privileged EXEC mode and returns to user EXEC mode.

Examples

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for Version 1 and Version 2 connections:

```
-----------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 1.99
```

```
Authentication timeout: 120 secs; Authentication retries: 3
-----------------------------------------------------------------------
```

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for a Version 2 connection with no Version 1 connection:

```
-----------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
-----------------------------------------------------------------------
```

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for a Version 1 connection with no Version 2 connection:

```
-----------------------------------------------------------------------
Device# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
-----------------------------------------------------------------------
```

Monitoring and Maintaining Secure Shell Version 2

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug ip ssh**

Example:

```
Device# debug ip ssh
```

Enables debugging of SSH.

Step 3 **debug snmp packet**

Example:

```
Device# debug snmp packet
```

Enables debugging of every SNMP packet sent or received by the device.

Example

The following sample output from the **debug ip ssh** command shows the connection is an SSH Version 2 connection:

```
Device# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
```

```
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
```

```
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally
```

Configuration Examples for Secure Shell Version 2 Support

Example: Configuring Secure Shell Version 1

```
Device# configure terminal
Device(config)# ip ssh version 1ip ssh version 2
```

Example: Configuring Secure Shell Version 2

```
Device# configure terminal
Device(config)# ip ssh version 2
```

Example: Configuring Secure Shell Versions 1 and 2

```
Device# configure terminal
Device(config)# no ip ssh version
```

Example: Starting an Encrypted Session with a Remote Device

```
Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

Example: Configuring Server-Side SCP

The following example shows how to configure the server-side functionality for SCP. This example also configures AAA authentication and authorization on the device. This example uses a locally defined username and password.

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username samplename privilege 15 password password1
Device(config)# ip ssh time-out 120
Device(config)# ip ssh authentication-retries 3
Device(config)# ip scp server enable
```

Example: Setting an SNMP Trap

The following example shows that an SNMP trap is set. The trap notification is generated automatically when the SSH session terminates. In the example, a.b.c.d is the IP address of the SSH client. For an example of SNMP trap debug output, see the Example: SNMP Debugging, on page 298 section.

```
snmp-server
snmp-server host a.b.c.d public tty
```

Examples: SSH Keyboard Interactive Authentication

Example: Enabling Client-Side Debugs

The following example shows that the client-side debugs are turned on, and the maximum number of prompts is six (three for the SSH keyboard interactive authentication method and three for the password authentication method).

```
Password:
Password:
Password:
Password:
Password:
Password: cisco123
Last login: Tue Dec 6 13:15:21 2005 from 10.76.248.213
user1@courier:~> exit
logout
[Connection to 10.76.248.200 closed by foreign host]
Device1# debug ip ssh client

SSH Client debugging is on

Device1# ssh -l lab 10.1.1.3
```

```
Password:
*Nov 17 12:50:53.199: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version exchange successful
*Nov 17 12:50:53.203: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.335: SSH CLIENT0: key exchange successful and encryption on
*Nov 17 12:50:53.335: SSH2 CLIENT 0: using method keyboard-interactive
Password:
Password:
Password:
*Nov 17 12:51:01.887: SSH2 CLIENT 0: using method password authentication
Password:
Password: lab
Device2>

*Nov 17 12:51:11.407: SSH2 CLIENT 0: SSH2_MSG_USERAUTH_SUCCESS message received
*Nov 17 12:51:11.407: SSH CLIENT0: user authenticated
*Nov 17 12:51:11.407: SSH2 CLIENT 0: pty-req request sent
*Nov 17 12:51:11.411: SSH2 CLIENT 0: shell request sent
*Nov 17 12:51:11.411: SSH CLIENT0: session open
```

Example: Enabling ChPass with a Blank Password Change

In the following example, the ChPass feature is enabled, and a blank password change is accomplished using the SSH Keyboard Interactive Authentication method. A TACACS+ access control server (ACS) is used as the back-end AAA server.

```
Device1# ssh -l cisco 10.1.1.3

Password:
Old Password: cisco
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]
```

Example: Enabling ChPass and Changing the Password on First Login

In the following example, the ChPass feature is enabled and TACACS+ ACS is used as the back-end server. The password is changed on the first login using the SSH keyboard interactive authentication method.

```
Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password:cisco1
```

```
Your password has expired.
Enter a new one now.
New Password: cisco
Re-enter New password: cisco12
The New and Re-entered passwords have to be the same.
Try again.
New Password: cisco
Re-enter New password: cisco

Device2>
```

Example: Enabling ChPass and Expiring the Password After Three Logins

In the following example, the ChPass feature is enabled and TACACS+ ACS is used as the back-end AAA server. The password expires after three logins using the SSH keyboard interactive authentication method.

```
Device# ssh -l cisco. 10.1.1.3

Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco

Device2> exit

Device1# ssh -l cisco 10.1.1.3

Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2>
```

Example: SNMP Debugging

The following is sample output from the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
Device1# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:
```

```
Device2# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Device1#
```

Examples: SSH Debugging Enhancements

The following is sample output from the **debug ip ssh detail** command. The output provides debugging information about the SSH protocol and channel requests.

```
Device# debug ip ssh detail

00:04:22: SSH0: starting SSH control process
00:04:22: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
00:04:22: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
00:04:22: SSH2 0: SSH2_MSG_KEXINIT sent
00:04:22: SSH2 0: SSH2_MSG_KEXINIT received
00:04:22: SSH2:kex: client->server enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2:kex: server->client enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2 0: expecting SSH2_MSG_KEXDH_INIT
00:04:22: SSH2 0: SSH2_MSG_KEXDH_INIT received
00:04:22: SSH2: kex_derive_keys complete
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS sent
00:04:22: SSH2 0: waiting for SSH2_MSG_NEWKEYS
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS received
00:04:24: SSH2 0: authentication successful for lab
00:04:24: SSH2 0: channel open request
00:04:24: SSH2 0: pty-req request
00:04:24: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width 80
00:04:24: SSH2 0: shell request
00:04:24: SSH2 0: shell message received
00:04:24: SSH2 0: starting shell for vty
00:04:38: SSH0: Session terminated normally
```

The following is sample output from the **debug ip ssh packet** command. The output provides debugging information about the SSH packet.

```
Device# debug ip ssh packet

00:05:43: SSH2 0: send:packet of  length 280 (length also includes padlen of 4)
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 280 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
```

```
00:05:43: SSH2 0: ssh_receive: 24 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 4 bytes
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 144 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 6 bytes
00:05:43: SSH2 0: signature length 143
00:05:43: SSH2 0: send:packet of  length 448 (length also includes padlen of 7)
00:05:43: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
00:05:43: SSH2 0: newkeys: mode 1
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: input: total packet length of 16 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 8 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 10 bytes
00:05:43: SSH2 0: newkeys: mode 0
00:05:43: SSH2 0: ssh_receive: 52 bytes received
00:05:43: SSH2 0: input: total packet length of 32 bytes
00:05:43: SSH2 0: partial packet length(block size)16 bytes,needed 16 bytes, maclen 20
00:05:43: SSH2 0: MAC compared for #3 :ok
```

Additional References for Secure Shell Version 2 Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
AAA Hostname and host domain configuration tasks Secure shell configuration tasks	*Security Configuration Guide: Securing User Services*
Downloading a software image Configuration fundamentals	*Configuration Fundamentals Configuration Guide*
IPsec configuration tasks	*Security Configuration Guide: Secure Connectivity*
SNMP traps configuration tasks	*SNMP Configuration Guide*

Standards

Standards	Title
IETF Secure Shell Version 2 Draft Standards	Internet Engineering Task Force website

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Secure Shell Version 2 Support

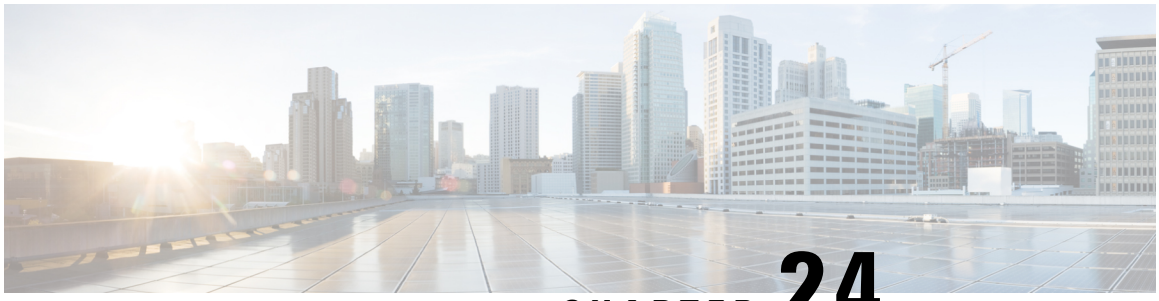
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 41: Feature Information for Secure Shell Version 2 Support

Feature Name	Releases	Feature Information
Secure Shell Version 2 Support		The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSH version 2 also supports AES counter-based encryption mode. The following commands were introduced or modified: **debug ip ssh**, **ip ssh min dh size**, **ip ssh rsa keypair-name**, **ip ssh version**, **ssh**.
Secure Shell Version 2 Client and Server Support		The Cisco IOS image was updated to provide for the automatic generation of SNMP traps when an SSH session terminates.
SSH Keyboard Interactive Authentication		The SSH Keyboard Interactive Authentication feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature.
Secure Shell Version 2 Enhancements		The Secure Shell Version 2 Enhancements feature includes a number of additional capabilities such as support for VRF-aware SSH, SSH debug enhancements, and DH Group 14 and Group 16 exchange support. The following commands were introduced or modified: **debug ip ssh**, **ip ssh dh min size**.

Feature Name	Releases	Feature Information
Secure Shell Version 2 Enhancements for RSA Keys.		The Secure Shell Version 2 Enhancements for RSA Keys feature includes a number of additional capabilities to support RSA key-based user authentication for SSH and SSH server host key storage and verification. The following commands were introduced or modified: **ip ssh pubkey-chain**, **ip ssh stricthostkeycheck**.



CHAPTER 24

Secure Shell—Configuring User Authentication Methods

The Secure Shell—Configuring User Authentication Methods feature helps configure the user authentication methods available in the Secure Shell (SSH) server.

Restrictions for Secure Shell—Configuring User Authentication Methods

Secure Shell (SSH) server and SSH client are supported on data encryption software (DES) (56-bit) and 3DES (168-bit) images only.

Information About Secure Shell—Configuring User Authentication Methods

Secure Shell User Authentication Overview

Secure Shell (SSH) enables an SSH client to make a secure, encrypted connection to a Cisco device (Cisco IOS SSH server). The SSH client uses the SSH protocol to provide device authentication and encryption.

The SSH server supports three types of user authentication methods and sends these authentication methods to the SSH client in the following predefined order:

- Public-key authentication method
- Keyboard-interactive authentication method

- Password authentication method

By default, all the user authentication methods are enabled. Use the **no ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**} command to disable any specific user authentication method so that the disabled method is not negotiated in the SSH user authentication protocol. This feature helps the SSH server offer any preferred user authentication method in an order different from the predefined order. The disabled user authentication method can be enabled using the **ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**} command.

As per RFC 4252 (The Secure Shell (SSH) Authentication Protocol), the public-key authentication method is mandatory. This feature enables the SSH server to override the RFC behavior and disable any SSH user authentication method, including public-key authentication.

For example, if the SSH server prefers the password authentication method, the SSH server can disable the public-key and keyboard-interactive authentication methods.

How to Configure Secure Shell—Configuring User Authentication Methods

Configuring User Authentication for the SSH Server

Perform this task to configure user authentication methods in the Secure Shell (SSH) server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**}
4. **ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**}
5. **default ip ssh server authenticate user**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	**no ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**}
	Example:
	`Device(config)# no ip ssh server authenticate user publickey`
	`%SSH:Publickey disabled.Overriding RFC`
		Disables a user authentication method in the Secure Shell (SSH) server.
		Note A warning message is displayed when the **no ip ssh server authenticate user publickey** command is used to disable public-key authentication. This command overrides the RFC 4252 (The Secure Shell (SSH) Authentication Protocol) behavior, which states that public-key authentication is mandatory.

Step 4	**ip ssh server authenticate user** {**publickey** | **keyboard** | **pasword**} **Example:** `Device(config)# ip ssh server authenticate user publickey`	Enables the disabled user authentication method in the SSH server.
Step 5	**default ip ssh server authenticate user** **Example:** `Device(config)# default ip ssh server authenticate user`	Returns to the default behavior in which all user authentication methods are enabled in the predefined order.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

- If the public-key-based authentication method is disabled using the **no ip ssh server authenticate user publickey** command, the RFC 4252 (The Secure Shell (SSH) Authentication Protocol) behavior in which public-key authentication is mandatory is overridden and the following warning message is displayed:

  ```
  %SSH:Publickey disabled.Overriding RFC
  ```

- If all three authentication methods are disabled, the following warning message is displayed:

  ```
  %SSH:No auth method configured.Incoming connection will be dropped
  ```

- In the event of an incoming SSH session request from the SSH client when all three user authentication methods are disabled on the SSH server, the connection request is dropped at the SSH server and a system log message is available in the following format:

  ```
  %SSH-3-NO_USERAUTH: No auth method configured for SSH Server. Incoming connection from <ip address> (tty = <ttynum>) dropped
  ```

Verifying User Authentication for the SSH Server

SUMMARY STEPS

1. **enable**
2. **show ip ssh**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 show ip ssh

Displays the version and configuration data for Secure Shell (SSH).

Example:

The following sample output from the **show ip ssh** command confirms that all three user authentication methods are enabled in the SSH server:

```
Device# show ip ssh

Authentication methods:publickey,keyboard-interactive,password
```

The following sample output from the **show ip ssh** command confirms that all three user authentication methods are disabled in the SSH server:

```
Device# show ip ssh

Authentication methods:NONE
```

Configuration Examples for Secure Shell—Configuring User Authentication Methods

Example: Disabling User Authentication Methods

The following example shows how to disable the public-key-based authentication and keyboard-based authentication methods, allowing the SSH client to connect to the SSH server using the password-based authentication method:

```
Device> enable
Device# configure terminal
Device(config)# no ip ssh server authenticate user publickey
%SSH:Publickey disabled.Overriding RFC
Device(config)# no ip ssh server authenticate user keyboard
Device(config)# exit
```

Example: Enabling User Authentication Methods

The following example shows how to enable the public-key-based authentication and keyboard-based authentication methods:

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server authenticate user publickey
Device(config)# ip ssh server authenticate user keyboard
Device(config)# exit
```

Example: Configuring Default User Authentication Methods

The following example shows how to return to the default behavior in which all three user authentication methods are enabled in the predefined order:

```
Device> enable
Device# configure terminal
Device(config)# default ip ssh server authenticate user
Device(config)# exit
```

Additional References for Secure Shell—Configuring User Authentication Methods

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
SSH configuration	*Secure Shell Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*
RFC 4253	*The Secure Shell (SSH) Transport Layer Protocol*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

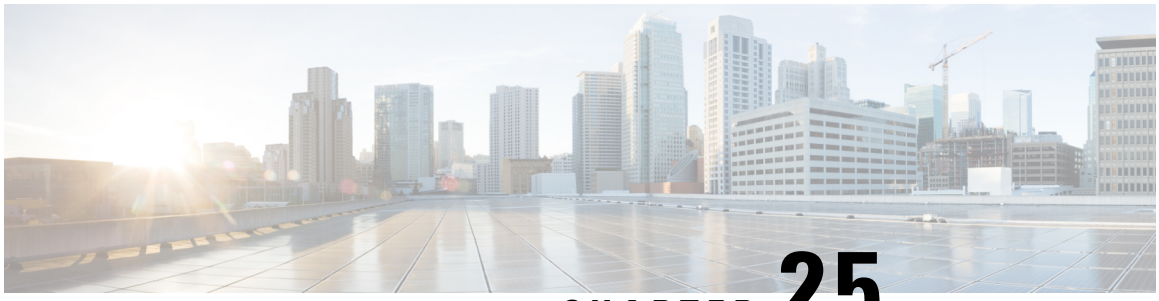
Feature Information for Secure Shell—Configuring User Authentication Methods

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 42: Feature Information for Secure Shell—Configuring User Authentication Methods

Feature Name	Releases	Feature Information
Secure Shell—Configuring User Authentication Methods	Cisco IOS XE Release 3.10S	The Secure Shell—Configuring User Authentication Methods feature helps configure the user authentication methods available in the Secure Shell (SSH) server. The following command was introduced: **ip ssh server authenticate user**. In Cisco IOS XE Release 3.10, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 25

X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for SSH Authentication feature uses the X.509v3 digital certificates in server and user authentication at the secure shell (SSH) server side.

This module describes how to configure server and user certificate profiles for a digital certificate.

Prerequisites for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature introduces the **ip ssh server algorithm authentication** command to replace the **ip ssh server authenticate user** command. If you use the **ip ssh server authenticate user** command, the following deprecation message is displayed.

```
Warning: SSH command accepted but this CLI will be deprecated soon. Please move to new
 CLI "ip ssh server algorithm authentication". Please configure "default ip ssh server
 authenticate user" to make CLI ineffective.
```

  - Use the **default ip ssh server authenticate user** command to remove the **ip ssh server authenticate user** command from effect. The IOS secure shell (SSH) server then starts using the **ip ssh server algorithm authentication** command.

Restrictions for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature implementation is applicable only on the IOS secure shell (SSH) server side.
- IOS SSH server supports only the x509v3-ssh-rsa algorithm based certificate for server and user authentication on the IOS SSH server side.

Information About X.509v3 Certificates for SSH Authentication

Digital certificates

The validity of the authentication depends upon the strength of the linkage between the public signing key and the identity of the signer. Digital certificates in the X.509v3 format (RFC5280) are used to provide identity management. A chain of signatures by a trusted root certification authority and its intermediate certificate authorities binds a given public signing key to a given digital identity.

Public key infrastructure (PKI) trustpoint helps manage the digital certificates. The association between the certificate and the trustpoint helps track the certificate. The trustpoint contains information about the certificate authority (CA), different identity parameters, and the digital certificate. Multiple trustpoints can be created to associate with different certificates.

Server and user authentication using X.509v3

For server authentication, the IOS secure shell (SSH) server sends its own certificate to the SSH client for verification. This server certificate is associated with the trustpoint configured in the server certificate profile (ssh-server-cert-profile-server configuration mode).

For user authentication, the SSH client sends the user's certificate to the IOS SSH server for verification. The SSH server validates the incoming user certificate using public key infrastructure (PKI) trustpoints configured in the server certificate profile (ssh-server-cert-profile-user configuration mode).

By default, certificate-based authentication is enabled for server and user at the IOS SSH server end.

How to Configure X.509v3 Certificates for SSH Authentication

Configuring IOS SSH Server to Use Digital Certificates for Sever Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}
4. **ip ssh server certificate profile**
5. **server**
6. **trustpoint sign** *PKI-trustpoint-name*
7. **ocsp-response include**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} **Example:** `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa`	Defines the order of host key algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client. **Note** The IOS SSH server must have at least one configured host key algorithm: • ssh-rsa – public key based authentication • x509v3-ssh-rsa – certificate-based authentication
Step 4	**ip ssh server certificate profile** **Example:** `Device(config)# ip ssh server certificate profile`	Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.
Step 5	**server** **Example:** `Device(ssh-server-cert-profile)# server`	Configures server certificate profile and enters SSH server certificate profile server configuration mode.
Step 6	**trustpoint sign** *PKI-trustpoint-name* **Example:** `Device(ssh-server-cert-profile-server)# trustpoint sign trust1`	Attaches the public key infrastructure (PKI) trustpoint to the server certificate profile. The SSH server uses the certificate associated with this PKI trustpoint for server authentication.
Step 7	**ocsp-response include** **Example:** `Device(ssh-server-cert-profile-server)# ocsp-response include`	(Optional) Sends the Online Certificate Status Protocol (OCSP) response or OCSP stapling along with the server certificate. **Note** By default the "no" form of this command is configured and no OCSP response is sent along with the server certificate.
Step 8	**end** **Example:** `Device(ssh-server-cert-profile-server)# end`	Exits SSH server certificate profile server configuration mode and enters privileged EXEC mode.

Configuring IOS SSH Server to Verify User's Digital Certificate for User Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**}
4. **ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}
5. **ip ssh server certificate profile**
6. **user**
7. **trustpoint verify** *PKI-trustpoint-name*
8. **ocsp-response required**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**} **Example:** `Device(config)# ip ssh server algorithm authentication publickey`	Defines the order of user authentication algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client. **Note** The IOS SSH server must have at least one configured user authentication algorithm. **Note** To use the certificate method for user authentication, the **publickey** keyword must be configured. **Note** The **ip ssh server algorithm authentication** command replaces the **ip ssh server authenticate user** command.
Step 4	**ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} **Example:**	Defines the order of public key algorithms. Only the configured algorithm is accepted by the SSH client for user authentication.

	Command or Action	Purpose
	`Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa`	**Note** The IOS SSH client must have at least one configured public key algorithm: • ssh-rsa – public-key-based authentication • x509v3-ssh-rsa – certificate-based authentication
Step 5	**ip ssh server certificate profile** **Example:** `Device(config)# ip ssh server certificate profile`	Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.
Step 6	**user** **Example:** `Device(ssh-server-cert-profile)# user`	Configures user certificate profile and enters SSH server certificate profile user configuration mode.
Step 7	**trustpoint verify** *PKI-trustpoint-name* **Example:** `Device(ssh-server-cert-profile-user)# trustpoint verify trust2`	Configures the public key infrastructure (PKI) trustpoint that is used to verify the incoming user certificate. **Note** Configure multiple trustpoints by executing the same command multiple times. A maximum of 10 trustpoints can be configured.
Step 8	**ocsp-response required** **Example:** `Device(ssh-server-cert-profile-user)# ocsp-response required`	(Optional) Mandates the presence of the Online Certificate Status Protocol (OCSP) response with the incoming user certificate. **Note** By default the "no" form of this command is configured and the user certificate is accepted without an OCSP response.
Step 9	**end** **Example:** `Device(ssh-server-cert-profile-user)# end`	Exits SSH server certificate profile user configuration mode and enters privileged EXEC mode.

Verifying Configuration for Server and User Authentication Using Digital Certificates

SUMMARY STEPS

1. **enable**
2. **show ip ssh**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show ip ssh**

Displays the currently configured authentication methods. To confirm the use of certificate-based authentication, ensure that the x509v3-ssh-rsa algorithm is the configured host key algorithm.

Example:

```
Device# show ip ssh

SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
```

Configuration Examples for X.509v3 Certificates for SSH Authentication

Example: Configuring IOS SSH Server to Use Digital Certificates for Sever Authentication

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# server
Device(ssh-server-cert-profile-server)# trustpoint sign trust1
Device(ssh-server-cert-profile-server)# exit
```

Example: Configuring IOS SSH Server to Verify User's Digital Certificate for User Authentication

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm authentication publickey
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# user
Device(ssh-server-cert-profile-user)# trustpoint verify trust2
Device(ssh-server-cert-profile-user)# end
```

Additional References for X.509v3 Certificates for SSH Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
SSH authentication	"Secure Shell-Configuring User Authentication Methods" chapter in *Secure Shell Configuration Guide*
Public key infrastructure (PKI) trustpoint	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" chapter in *Public Key Infrastructure Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

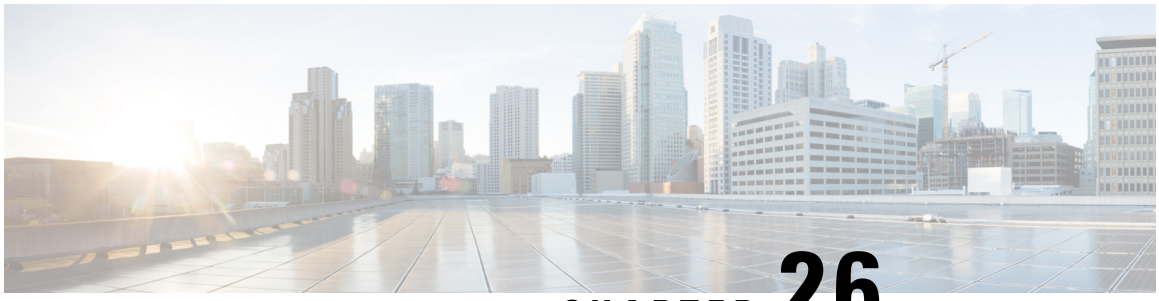
Feature Information for X.509v3 Certificates for SSH Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 43: Feature Information for X.509v3 Certificates for SSH Authentication

Feature Name	Releases	Feature Information
X.509v3 Certificates for SSH Authentication		The X.509v3 Certificates for SSH Authentication feature uses the X.509v3 digital certificates in server and user authentication at the secure shell (SSH) server side. The following commands were introduced or modified: **ip ssh server algorithm hostkey**, **ip ssh server algorithm authentication**, and **ip ssh server certificate profile**.



CHAPTER 26

SSH Algorithms for Common Criteria Certification

The SSH Algorithms for Common Criteria Certification feature provides the list and order of the algorithms that are allowed for Common Criteria Certification. This module describes how to configure the encryption, Message Authentication Code (MAC), and host key algorithms for a secure shell (SSH) server and client so that SSH connections can be limited on the basis of the allowed algorithms list.

Restriction for SSH Algorithms for Common Criteria Certification

- Starting from Cisco IOS XE Release 17.10, the following Key Exchange and MAC algorithms are removed from the default list:

  Key Exchange algorithm:

  - diffie-hellman-group14-sha1

  MAC algorithms:

  - hmac-sha1
  - hmac-sha2-256
  - hmac-sha2-512

Note You can use the **ip ssh server algorithm kex** command to configure the Key Exchange algorithm and the **ip ssh server algorithm mac** command to configure the MAC algorithms.

Information About SSH Algorithms for Common Criteria Certification

SSH Algorithms for Common Criteria Certification

A Secure Shell (SSH) configuration enables a Cisco IOS SSH server and client to authorize the negotiation of only those algorithms that are configured from the allowed list. If a remote party tries to negotiate using only those algorithms that are not part of the allowed list, the request is rejected and the session is not established.

Cisco IOS SSH Server Algorithms

Cisco IOS secure shell (SSH) servers support the encryption algorithms (Advanced Encryption Standard Counter Mode [AES-CTR], AES Cipher Block Chaining [AES-CBC], Triple Data Encryption Standard [3DES]), and Galois/Counter Mode (GCM)), the Message Authentication Code (MAC) algorithms, the host key algorithms, the Key Exchange (KEX) DH Group algorithms, and the public key algorithms in the following order:

Table 44: Supported Default and Non-Default IOS SSH Server Algorithms

Supported Algorithms	Default	Non-Default
Encryption	**1.** chacha20-poly1305@openssh.com **2.** aes128-gcm@openssh.com **3.** aes256-gcm@openssh.com **4.** aes128-gcm **5.** aes256-gcm **6.** aes128-ctr **7.** aes192-ctr **8.** aes256-ctr	• aes128-cbc • aes192-cbc • aes256-cbc • 3des-cbc
HMAC	**1.** hmac-sha2-256-etm@openssh.com **2.** hmac-sha2-512-etm@openssh.com	• hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
Host Key	**1.** rsa-sha2-512 **2.** rsa-sha2-256 **3.** ssh-rsa	• x509v3-ssh-rsa

Supported Algorithms	Default	Non-Default
KEX DH Group	**1.** curve25519-sha256 **2.** curve25519-sha256@libssh.org **3.** ecdh-sha2-nistp256 **4.** ecdh-sha2-nistp384 **5.** ecdh-sha2-nistp521 **6.** diffie-hellman-group14-sha256 **7.** diffie-hellman-group16-sha512	• diffie-hellman-group14-sha1
Public Key	**1.** ssh-rsa **2.** ecdsa-sha2-nistp256 **3.** ecdsa-sha2-nistp384 **4.** ecdsa-sha2-nistp521 **5.** ssh-ed25519 **6.** x509v3-ecdsa-sha2-nistp256 **7.** x509v3-ecdsa-sha2-nistp384 **8.** x509v3-ecdsa-sha2-nistp521 **9.** rsa-sha2-256 **10.** rsa-sha2-512 **11.** x509v3-rsa2048-sha256	• x509v3-ssh-rsa

Cisco IOS SSH Client Algorithms

Cisco IOS secure shell (SSH) clients support the encryption algorithms (Advanced Encryption Standard counter mode [AES-CTR], AES Cipher Block Chaining [AES-CBC], Triple Data Encryption Standard [3DES]), and Galois/Counter Mode (GCM)), the MAC algorithms, and the KEX DH Group algorithms in the following order:

Table 45: Supported Default and Non-Default IOS SSH Server Algorithms

Supported Algorithms	Default	Non-Default
Encryption	1. chacha20-poly1305@openssh.com 2. aes128-gcm@openssh.com 3. aes256-gcm@openssh.com 4. aes128-gcm 5. aes256-gcm 6. aes128-ctr 7. aes192-ctr 8. aes256-ctr	• aes128-cbc • aes192-cbc • aes256-cbc • 3des-cbc
HMAC	1. hmac-sha2-256-etm@openssh.com 2. hmac-sha2-512-etm@openssh.com	• hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
KEX DH Group	1. curve25519-sha256 2. curve25519-sha256@libssh.org 3. ecdh-sha2-nistp256 4. ecdh-sha2-nistp384 5. ecdh-sha2-nistp521 6. diffie-hellman-group14-sha256 7. diffie-hellman-group16-sha512	• diffie-hellman-group14-sha1

How to Configure SSH Algorithms for Common Criteria Certification

Configuring an Encryption Key Algorithm for a Cisco IOS SSH Server and Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh** {**server** | **client**} **algorithm encryption** {**aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-cbc** | **3des-cbc** | **aes192-cbc** | **aes256-cbc**}

4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip ssh** {**server** | **client**} **algorithm encryption** {**aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-cbc** | **3des-cbc** | **aes192-cbc** | **aes256-cbc**}
	Example:
	`Device(config)# ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc`
	`Device(config)# ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc`
		Defines the order of encryption algorithms in the SSH server and client. This order is presented during algorithm negotiation.
		Note The Cisco IOS SSH server and client must have at least one configured encryption algorithm.
		Note To disable one algorithm from the previously configured algorithm list, use the **no** form of this command. To disable more than one algorithm, use the **no** form of this command multiple times with different algorithm names.
		Note For a default configuration, use the default form of this command as shown below:
		`Device(config)# ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc`

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you try to disable the last encryption algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All encryption algorithms cannot be disabled
```

Configuring a MAC Algorithm for a Cisco IOS SSH Server and Client

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip ssh** {**server** | **client**} **algorithm mac** {**hmac-sha2** | **hmac-sha2-96**}
	Example:
	`Device(config)# ip ssh server algorithm mac hmac-sha2 hmac-sha2-96`
	`Device(config)# ip ssh client algorithm mac hmac-sha2 hmac-sha2-96`
		Defines the order of MAC (Message Authentication Code) algorithms in the SSH server and client. This order is presented during algorithm negotiation.
		Note The Cisco IOS SSH server and client must have at least one configured Hashed Message Authentication Code (HMAC) algorithm.
		Note To disable one algorithm from the previously configured algorithm list, use the **no** form of this command. To disable more than one algorithm, use the **no** form of this command multiple times with different algorithm names.
		Note For default configuration, use the default form of this command as shown below:
		`Device(config)# ip ssh server algorithm mac hmac-sha2 hmac-sha2-96`

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you try to disable the last MAC algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All mac algorithms cannot be disabled
```

Configuring a Host Key Algorithm for a Cisco IOS SSH Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** | **ssh-rsa**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh server algorithm hostkey** {**x509v3-ssh-rsa**	**ssh-rsa**} **Example:** `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa`	Defines the order of host key algorithms. Only the configured algorithm is negotiated with the Cisco IOS secure shell (SSH) client. **Note** The Cisco IOS SSH server must have at least one configured host key algorithm: • x509v3-ssh-rsa—X.509v3 certificate-based authentication • ssh-rsa—Public-key-based authentication **Note** To disable one algorithm from the previously configured algorithm list, use the **no** form of this command. To disable more than one algorithm, use the **no** form of this command multiple times with different algorithm names. **Note** For default configuration, use the default form of this command as shown below: `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa`
Step 4	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Troubleshooting Tips

If you try to disable the last host key algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All hostkey algorithms cannot be disabled
```

Verifying SSH Algorithms for Common Criteria Certification

SUMMARY STEPS

1. **enable**
2. **show ip ssh**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show ip ssh**

Displays configured Secure Shell (SSH) encryption, host key, and Message Authentication Code (MAC) algorithms.

Example:

The following sample output from the **show ip ssh** command shows the encryption algorithms configured in the default order:

```
Device# show ip ssh
Encryption Algorithms: aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, 3des-cbc, aes192-cbc, aes256-cbc
```

The following sample output from the **show ip ssh** command shows the MAC algorithms configured in the default order:

```
Device# show ip ssh
MAC Algorithms: hmac-sha1 hmac-sha1-96
```

The following sample output from the **show ip ssh** command shows the host key algorithms configured in the default order:

```
Device# show ip ssh

Hostkey Algorithms: x509v3-ssh-rsa, ssh-rsa
```

Configuration Examples for SSH Algorithms for Common Criteria Certification

Example: Configuring Encryption Key Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc
 3des-cbc aes192-cbc aes256-cbc
Device(config)# end
```

Example: Configuring Encryption Key Algorithms for a Cisco IOS SSH Client

```
Device> enable
Device# configure terminal
Device(config)# ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc
 3des-cbc aes192-cbc aes256-cbc
Device(config)# end
```

Example: Configuring MAC Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm mac hmac-sha1 hmac-sha1-96
Device(config)# end
```

Example: Configuring Key Exchange DH Group for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
```

```
Device(config)# ip ssh server algorithm kex diffie-hellman-group-exchange-sha1
Device(config)# end
```

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm kex diffie-hellman-group14-sha1
Device(config)# end
```

Example: Configuring Host Key Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa
Device(config)# end
```

Additional References for SSH Algorithms for Common Criteria Certification

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
SSH authentication	"Secure Shell-Configuring User Authentication Methods" chapter in the *Secure Shell Configuration Guide*
X.509v3 digital certificates in server and user authentication	"X.509v3 Certificates for SSH Authentication" chapter in the *Secure Shell Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSH Algorithms for Common Criteria Certification

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for SSH Algorithms for Common Criteria Certification

Feature Name	Releases	Feature Information
SSH Algorithms for Common Criteria Certification	Cisco IOS XE Everest 16.5.1a	The SSH Algorithms for Common Criteria Certification feature provides the list and order of the algorithms that are allowed for Common Criteria Certification. This module describes how to configure the encryption, Message Authentication Code (MAC), and host key algorithms for a secure shell (SSH) server and client so that SSH connections can be limited on the basis of the allowed algorithms list. The following commands were introduced by this feature: **ip ssh** {**server** | **client**} **algorithm encryption**, **ip ssh** {**server** | **client**} **algorithm mac**.
SSH Algorithms for Common Criteria Certification	Cisco IOS XE Cupertino 17.8.1	Cisco IOS SSH Server and Client support for the following algorithms have been introduced: • chacha20-poly1305@openssh.com • ssh-ed25519 • curve25519-sha256@libssh.org

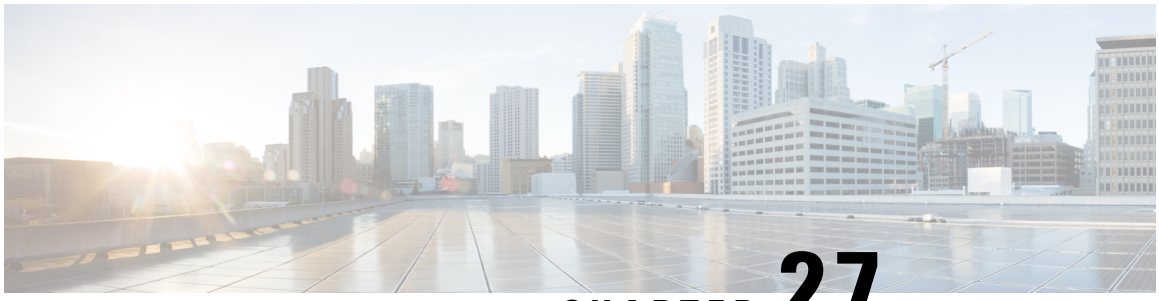
Feature Name	Releases	Feature Information
SSH Algorithms for Common Criteria Certification	Cisco IOS XE Cupertino 17.9.1	Cisco IOS SSH Server and Client support for the following algorithms have been introduced: • aes128-gcm@openssh.com • aes256-gcm@openssh.com

Deprecation of Weak Ciphers	Cisco IOS XE Release 17.10	The following changes have been introduced:
		• The Secure Shell Version 1.99 is not supported.
		• The following weak Key Exchange and MAC algorithms are removed from the default list of algorithms:
		  • diffie-hellman-group14-sha1
		  • hmac-sha1
		  • hmac-sha2-256
		  • hmac-sha2-512

SSH Algorithms for Common Criteria Certification	Cisco IOS XE Release 17.11.1a	Cisco IOS SSH Server and Client support for the following algorithms have been introduced: • curve25519-sha256 • diffie-hellman-group14-sha256 • diffie-hellman-group16-sha512 • x509v3-rsa2048-sha256

PART III

Access Control Lists



CHAPTER 27

IP Access List Overview

Access control lists (ACLs) perform packet filtering to control which packets move through a network and to where. The packet filtering provides security by helping to limit the network traffic, restrict the access of users and devices to a network, and prevent the traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as bandwidth control, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features. This module provides an overview of IP access lists.

Information About IP Access Lists

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.
- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.

- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.
- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Border Routers and Firewall Routers Should Use Access Lists

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide a basic level of security for your network by controlling access to it. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

An access list can allow one host to access a part of your network and prevent another host from accessing the same area. In the figure below, by applying an appropriate access list to the interfaces of the router, Host A is allowed to access the Human Resources network and Host B is prevented from accessing the Human Resources network.

Access lists should be used in firewall routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide some security benefits of access lists, you should at least configure access lists on border routers--routers located at the edges of your networks. Such an access list provides a basic buffer from the outside network or from a less controlled area of your own network into a more sensitive area of your network. On these border routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists are defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Definition of an Access List

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets.

Access lists are identified and referenced by a name or a number. Access lists act as packet filters, filtering packets based on the criteria defined in each access list.

After you configure an access list, for the access list to take effect, you must either apply the access list to an interface (by using the **ip access-group** command), a vty (by using the **access-class** command), or reference the access list by any command that accepts an access list. Multiple commands can reference the same access list.

In the following configuration, an IP access list named branchoffices is configured on Fast Ethernet interface 0/1/0 and applied to incoming packets. Networks other than the ones specified by the source address and mask pair cannot access Fast Ethernet interface 0/1/0. The destinations for packets coming from sources on network 172.16.7.0 are unrestricted. The destination for packets coming from sources on network 172.16.2.0 must be 172.31.5.4.

```
ip access-list extended branchoffices
 10 permit 172.16.7.0 0.0.0.3 any
 20 permit 172.16.2.0 0.0.0.255 host 172.31.5.4
!
interface fastethernet 0/1/0
 ip access-group branchoffices in
```

Access List Rules

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you

configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.

- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

AccessList Rules for Dialer Lists

The following access list rules apply only to Cisco ISR 4000 series platforms:

- The dialer interface on serial interface (BRI/PRI) uses egress ACL to dial out. Therefore, the ACL configuration for dialer list must be egress ACL.
- Dialer idle-timeout must be configured with outbound direction. Inbound dialer idle-timeout configuration with ingress ACL list for dialer list will cause session to idle timeout.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.

- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a task. You can reorder statements in or add statements to a named access list.

Named access lists support the following features that are not supported by numbered access lists:

- IP options filtering
- Noncontiguous ports
- TCP flag filtering
- Deleting of entries with the **no permit** or **no deny** command

Note Not all commands that accept a numbered access list will accept a named access list. For example, vty uses only numbered access lists.

Standard or Extended Access Lists

All access lists are either standard or extended access lists. If you only intend to filter on a source address, the simpler standard access list is sufficient. For filtering on anything other than a source address, an extended access list is necessary.

- Named access lists are specified as standard or extended based on the keyword **standard** or **extended** in the **ip access-list** command syntax.
- Numbered access lists are specified as standard or extended based on their number in the **access-list** command syntax. Standard IP access lists are numbered 1 to 99 or 1300 to 1999; extended IP access lists are numbered 100 to 199 or 2000 to 2699. The range of standard IP access lists was initially only 1 to 99, and was subsequently expanded with the range 1300 to 1999 (the intervening numbers were assigned to other protocols). The extended access list range was similarly expanded.

Note Starting from Cisco IOS XE 16.9.4, use the **ip access-list** command to configure object-group based numbered ACL.

Standard Access Lists

Standard IP access lists test only source addresses of packets (except for two exceptions). Because standard access lists test source addresses, they are very efficient at blocking traffic close to a destination. There are two exceptions when the address in a standard access list is not a source address:

- On outbound VTY access lists, when someone is trying to telnet, the address in the access list entry is used as a destination address rather than a source address.
- When filtering routes, you are filtering the network being advertised to you rather than a source address.

Extended Access Lists

Extended access lists are good for blocking traffic anywhere. Extended access lists test source and destination addresses and other IP packet data, such as protocols, TCP or UDP port numbers, type of service (ToS), precedence, TCP flags, and IP options. Extended access lists can also provide capabilities that standard access lists cannot, such as the following:

- Filtering IP Options
- Filtering TCP flags
- Filtering noninitial fragments of packets (see the module "Refining an IP Access List")

Note Packets that are subject to an extended access list will not be autonomous switched.

IP Packet Fields You Can Filter to Control Access

You can use an extended access list to filter on any of the following fields in an IP packet. Source address and destination address are the two most frequently specified fields on which to base an access list:

- Source address--Specifies a source address to control packets coming from certain networking devices or hosts.
- Destination address--Specifies a destination address to control packets being sent to certain networking devices or hosts.
- Protocol--Specifies an IP protocol indicated by the keyword **eigrp**, **gre**, **icmp**, **igmp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or indicated by an integer in the range from 0 to 255 (representing an Internet protocol). If you specify a transport layer protocol (**icmp**, **igmp**, **tcp**, or **udp**), the command has a specific syntax.
  - Ports and non-contiguous ports--Specifies TCP or UDP ports by a port name or port number. The port numbers can be noncontiguous port numbers. Port numbers can be useful to filter Telnet traffic or HTTP traffic, for example.
  - TCP flags--Specifies that packets match any flag or all flags set in TCP packets. Filtering on specific TCP flags can help prevent false synchronization packets.
- IP options--Specifies IP options; one reason to filter on IP options is to prevent routers from being saturated with spurious packets containing them.

Wildcard Mask for Addresses in an Access List

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, you can specify one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value; they must match.
- A wildcard mask bit 1 means ignore that corresponding bit value; they need not match.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes an implicit wildcard mask of 0.0.0.0, meaning all values must match.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

The table below shows examples of IP addresses and masks from an access list, along with the corresponding addresses that are considered a match.

Table 47: Sample IP Addresses, Wildcard Masks, and Match Results

Address	Wildcard Mask	Match Results
0.0.0.0	255.255.255.255	All addresses will match the access list conditions.
172.18.0.0/16	0.0.255.255	Network 172.18.0.0
172.18.5.2/16	0.0.0.0	Only host 172.18.5.2 matches
172.18.8.0	0.0.0.7	Only subnet 172.18.8.0/29 matches
172.18.8.8	0.0.0.7	Only subnet 172.18.8.8/29 matches
172.18.8.15	0.0.0.3	Only subnet 172.18.8.15/30 matches
10.1.2.0	0.0.252.255 (noncontiguous bits in mask)	Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

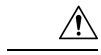
This feature allows users to add sequence numbers to access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Access List Logging

The Cisco IOS software can provide logging messages about packets permitted or denied by a single standard or extended IP access list entry. That is, any packet that matches the entry will cause an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the **logging console** global configuration command.

The first packet that triggers the access list entry causes an immediate logging message, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

However, you can use the **ip access-list log-update** command to set the number of packets that, when match an access list (and are permitted or denied), cause the system to generate a log message. You might want to do this to receive log messages more frequently than at 5-minute intervals.



Caution If you set the *number-of-matches* argument to 1, a log message is sent right away, rather than caching it; every packet that matches an access list causes a log message. A setting of 1 is not recommended because the volume of log messages could overwhelm the system.

Even if you use the **ip access-list log-update** command, the 5-minute timer remains in effect, so each cache is emptied at the end of 5 minutes, regardless of the count of messages in each cache. Regardless of when the log message is sent, the cache is flushed and the count reset to 0 for that message the same way it is when a threshold is not specified.

Note The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

Alternative to Access List Logging

Packets matching an entry in an ACL with a log option are process switched. It is not recommended to use the log option on ACLs, but rather use NetFlow export and match on a destination interface of Null0. This is done in the CEF path. The destination interface of Null0 is set for any packet that is dropped by the ACL.

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the module entitled “Refining an Access List.”

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.
- After you create a named access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

RSP3 Porting Related Information

Outbound Access List is not supported in RSP3.

Where to Apply an Access List

You can apply access lists to the inbound or outbound interfaces of a device. Applying an access list to an inbound interface controls the traffic that enters the interface and applying an access list to an outbound interface controls the traffic that exits the interface.

When software receives a packet at the inbound interface, the software checks the packet against the statements that are configured for the access list. If the access list permits packets, the software processes the packet. Applying access lists to filter incoming packets can save device resources because filtered packets are discarded before entering the device.

Access lists on outbound interfaces filter packets that are transmitted (sent) out of the interface. You can use the TCP Access Control List (ACL) Splitting feature of the Rate-Based Satellite Control Protocol (RBSCP) on the outbound interface to control the type of packets that are subject to TCP acknowledgment (ACK) splitting on an outbound interface.

You can reference an access list by using a **debug** command to limit the amount of debug logs. For example, based on the filtering or matching criteria of the access list, debug logs can be limited to source or destination addresses or protocols.

You can use access lists to control routing updates, dial-on-demand (DDR), and quality of service (QoS) features.

Additional References

Related Documents

Related Topic	Document Title
IP access list commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Filtering on source address, destination address, or protocol	Creating an IP Access List and Applying It to an Interface” module
Filtering on IP Options, TCP flags, noncontiguous ports, or TTL	Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports module

Standards

Standards & RFCs	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

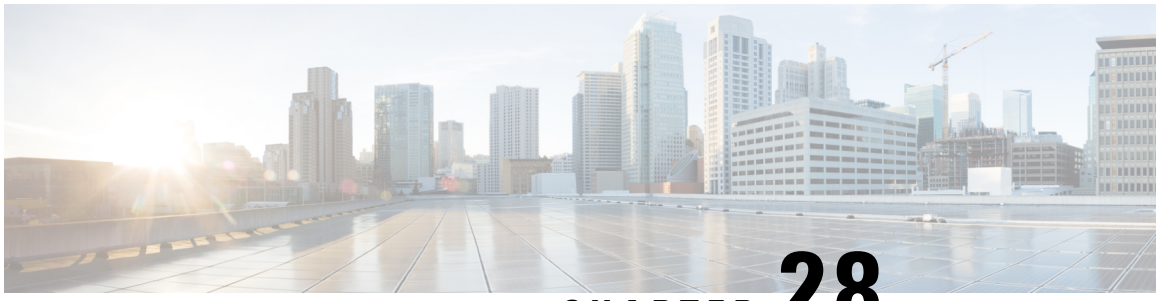
Feature Information for IP Access Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 48: Feature Information for IP Access Lists

Feature Name	Releases	Feature Configuration Information
ACL—IP Protocol	Cisco IOS XE Release 3.16	In Cisco IOS XE Release 3.16, support was added for the Cisco ASR 903 Router.



CHAPTER 28

Creating an IP Access List and Applying It to an Interface

IP access lists provide many benefits for securing a network and achieving nonsecurity goals, such as determining quality of service (QoS) factors or limiting **debug** command output. This module describes how to create standard, extended, named, and numbered IP access lists. An access list can be referenced by a name or a number. Standard access lists filter on only the source address in IP packets. Extended access lists can filter on source address, destination address, and other fields in an IP packet.

After you create an access list, you must apply it to something in order for it to have any effect. This module describes how to apply an access list to an interface. However, there are many other uses for access lists, which are mentioned in this module and described in other modules and in other configuration guides for various technologies.

Restrictions for Creating an IP Access List and Applying It to an Interface

The following restrictions apply when configuring IPv4 and IPv6 access control lists (ACLs)

- Application control engine (ACE)-specific counters are not supported.
- Layer 3 IPv4 and IPv6 ACLs are not supported on the same interface.
- MAC ACLs are not supported on Ethernet flow points (EFPs) or trunk EFP interfaces to which Layer 3 IPv4 or IPv6 ACLs are applied.
- IPv4 and IPv6 ACLs are not currently supported on EFP interfaces. IPv4 and IPv6 ACLs are supported on physical interfaces, bridge-domain interfaces, and port-channel interfaces.

- Layer 4 port-range functionality expands into Ternary Content-Addressable Memory (TCAM). IPv4 ACL scale is limited to 1K TCAM, Layer 2 ACL scale is limited to 1K TCAM entries.
- Object-groups ACLs (IPv4 and IPv6 ACLs) are supported on Cisco ISR platforms.
- The command **any options** is not supoprted.
- Starting with Cisco IOS XE Cupertino Release 17.7.1, ACLs are supported on management interface, Gigabit 0.

Information About Creating an IP Access List and Applying It to an Interface

Helpful Hints for Creating IP Access Lists

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- A packet will match the first ACE in the ACL. Thus, a **permit ip any any** will match all packets, ignoring all subsequent ACES.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry. You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
 remark Do not allow host1 subnet to telnet out
 deny tcp host 172.16.2.88 any eq telnet
```

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the *Refining an IP Access List module*.

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.
- After you create a named or numbered access list, you might want to add entries or change the order of the entries, which are known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

How to Create an IP Access List and Apply It to an Interface

This section describes the general ways to create a standard or extended access list using either a name or a number. Access lists are very flexible; the tasks simply illustrate one **permit** command and one **deny** command to provide you the command syntax of each. Only you can determine how many **permit** and **deny** commands you need and their order.

Note The first two tasks in this module create an access list; you must apply the access list in order for it to function. If you want to apply the access list to an interface, perform the task "Applying the Access List to an Interface".

Creating a Standard Access List to Filter on Source Address

If you want to filter on source address only, a standard access list is simple and sufficient. There are two alternative types of standard access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features than numbered access lists.

Creating a Named Access List to Filter on Source Address

Use a standard, named access list if you need to filter on source address only. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip access-list standard** *name*

Example:

```
Device(config)# ip access-list standard R&D
```

Defines a standard IP access list using a name and enters standard named access list configuration mode.

Step 4 **remark** *remark*

Example:

```
Device(config-std-nacl)# remark deny Sales network
```

(Optional) Adds a user-friendly comment about an access list entry.

- A remark can precede or follow an access list entry.
- In this example, the remark reminds the network administrator that the subsequent entry denies the Sales network access to the interface (assuming this access list is later applied to an interface).

Step 5 **deny** {*source* [*source-wildcard*] | **any**} [**log**]

Example:

```
Device(config-std-nacl)# deny 172.16.0.0 0.0.255.255 log
```

(Optional) Denies the specified source based on a source address and wildcard mask.

- If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.
- Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255.

- In this example, all hosts on network 172.16.0.0 are denied passing the access list.
- Because this example explicitly denies a source address and the **log** keyword is specified, any packets from that source are logged when they are denied. This is a way to be notified that someone on a network or host is trying to gain access.

Step 6 **remark** *remark*

Example:

```
Device(config-std-nacl)# remark Give access to Tester's host
```

(Optional) Adds a user-friendly comment about an access list entry.

- A remark can precede or follow an access list entry.
- This remark reminds the network administrator that the subsequent entry allows the Tester's host access to the interface.

Step 7 **permit** {*source* [*source-wildcard*] | **any**} [**log**]

Example:

```
Device(config-std-nacl)# permit 172.18.5.22 0.0.0.0
```

Permits the specified source based on a source address and wildcard mask.

- Every access list needs at least one **permit** statement; it need not be the first entry.
- If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.
- Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255.
- In this example, host 172.18.5.22 is allowed to pass the access list.

Step 8 Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.

Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.

Step 9 **end**

Example:

```
Device(config-std-nacl)# end
```

Exits standard named access list configuration mode and enters privileged EXEC mode.

Step 10 **show ip access-list**

Example:

```
Device# show ip access-list
```

(Optional) Displays the contents of all current IP access lists.

Creating a Numbered Access List to Filter on Source Address

Configure a standard, numbered access list if you need to filter on source address only and you prefer not to use a named access list.

IP standard access lists are numbered 1 to 99 or 1300 to 1999. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **access-list** *access-list-number permit* {*source* [*source-wildcard*] | **any**} [**log**]

Example:

```
Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0
```

Permits the specified source based on a source address and wildcard mask.

- Every access list needs at least one permit statement; it need not be the first entry.
- Standard IP access lists are numbered 1 to 99 or 1300 to 1999.
- If the source-wildcard is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.
- Optionally use the keyword any as a substitute for the source source-wildcard to specify the source and source wildcard of 0.0.0.0 255.255.255.255.
- In this example, host 172.16.5.22 is allowed to pass the access list.

Step 4 **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]

Example:

```
Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0
```

Denies the specified source based on a source address and wildcard mask.

- If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.
- Optionally use the abbreviation **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255.
- In this example, host 172.16.7.34 is denied passing the access list.

Step 5 Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.

Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.

Step 6 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and enters privileged EXEC mode.

Step 7 **show ip access-list**

Example:

```
Device# show ip access-list
```

(Optional) Displays the contents of all current IP access lists.

Creating an Extended Access List

If you want to filter on anything other than source address, you need to create an extended access list. There are two alternative types of extended access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features.

For details on how to filter something other than source or destination address, see the syntax descriptions in the command reference documentation.

Creating a Named Extended Access List

Create a named extended access list if you want to filter the source and destination address or filter a combination of addresses and other IP fields.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*

4. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.
7. **end**
8. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended acl1`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**	**log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# deny ip 172.18.0.0 0.0.255.255 host 172.16.40.10 log`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • Optionally use the keyword **host** *source* to indicate a source and source wildcard of *source* 0.0.0.0 or the abbreviation **host** *destination* to indicate a destination and destination wildcard of *destination* 0.0.0.0. • In this example, packets from all sources are denied access to the destination network 172.18.0.0. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the

	Command or Action	Purpose
		logging facility command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the **logging console** command.
Step 5	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one permit statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • In this example, TCP packets are allowed from any source to any destination. • Use the **log-input** keyword to include input interface, source MAC address, or virtual circuit in the logging output.
Step 6	Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 7	**end** **Example:** `Device(config-ext-nacl)# end`	Exits standard named access list configuration mode and enters privileged EXEC mode.
Step 8	**show ip access-list** **Example:** `Device# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

RSP3 Porting Related Information

ACL is not supported for fragmented packets.

Creating a Numbered Extended Access List

Create a numbered extended access list if you want to filter on source and destination address, or a combination of addresses and other IP fields, and you prefer not to use a name. Extended IP access lists are numbered 100 to 199 or 2000 to 2699.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.
8. **end**
9. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 107 remark allow Telnet packets from any source to network 172.69.0.0 (headquarters)`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*]	**any**} {*destination* [*destination-wildcard*]	**any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**	**log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:**	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement; it need not be the first entry.

	Command or Action	Purpose
	`Device(config)# access-list 107 permit tcp any 172.69.0.0 0.0.255.255 eq telnet`	• Extended IP access lists are numbered 100 to 199 or 2000 to 2699. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • TCP and other protocols have additional syntax available. See the **access-list** command in the command reference for complete syntax.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 107 remark deny all other TCP packets`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 6	**access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config)# access-list 107 deny tcp any any`	Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 9	**show ip access-list** **Example:** `Device# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Applying an Access List to a Physical Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
5. **ip access-list extended** *acl-name acl-number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Device(config-if)# ip access-group acl1 in`	Applies the specified access list to the inbound interface. • To filter source addresses, apply the access list to the inbound interface.
Step 5	**ip access-list extended** *acl-name acl-number* **Example:**	Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list. Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list. • Access list to which all commands entered from ACL configuration mode apply, using an alphanumeric string of up to 30 characters, beginning with a letter. • Access list to which all commands entered from access list configuration mode apply, using a numeric identifier. For extended access lists, the valid range is 100 to 199.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Creating an IP Access List and Applying It to a Physical Interface

Example: Filtering on Host Source Address

In the following example, the workstation belonging to user1 is allowed access to gigabitethernet 0/0/0, and the workstation belonging to user2 is not allowed access:

```
interface gigabitethernet 0/0/0
 ip access-group workstations in
!
ip access-list standard workstations
 remark Permit only user1 workstation through
 permit 172.16.2.88
 remark Do not allow user2 workstation through
 deny 172.16.3.13
```

Example: Filtering on Subnet Source Address

In the following example, the user1 subnet is not allowed access to gigabitethernet interface 0/0/0, but the Main subnet is allowed access:

```
interface gigabitethernet 0/0/0
 ip access-group prevention in
!
ip access-list standard prevention
 remark Do not allow user1 subnet through
 deny 172.22.0.0 0.0.255.255
 remark Allow Main subnet
 permit 172.25.0.0 0.0.255.255
```

Example: Filtering on Source and Destination Addresses and IP Protocols

The following configuration example shows an interface with two access lists, one applied to outgoing packets and one applied to incoming packets. The standard access list named Internet-filter filters outgoing packets on source address. The only packets allowed out the interface must be from source 172.16.3.4.

The extended access list named marketing-group filters incoming packets. The access list permits Telnet packets from any source to network 172.26.0.0 and denies all other TCP packets. It permits any ICMP packets. It denies UDP packets from any source to network 172.26.0 0 on port numbers less than 1024. Finally, the access list denies all other IP packets and performs logging of packets passed or denied by that entry.

```
interface gigabitethernet 0/0/0
 ip address 172.20.5.1 255.255.255.0
 ip access-group Internet-filter out
 ip access-group marketing-group in
!
ip access-list standard Internet-filter
 permit 172.16.3.4
ip access-list extended marketing-group
 permit tcp any 172.26.0.0 0.0.255.255 eq telnet
 deny tcp any any
 permit icmp any any
 deny udp any 172.26.0.0 0.0.255.255 lt 1024
 deny ip any any
```

Example: Filtering on Source Addresses Using a Numbered Access List

In the following example, network 10.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 10.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS XE software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 10.0.0.0 subnets.

```
interface gigabitethernet 0/0/0
 ip access-group 2 in
!
access-list 2 permit 10.48.0.3
access-list 2 deny 10.48.0.0  0.0.255.255
access-list 2 permit 10.0.0.0  0.255.255.255
```

Example: Preventing Telnet Access to a Subnet

In the following example, the user1 subnet is not allowed to telnet out of gigabitethernet interface 0/0/0:

```
interface gigabitethernet 0/0/0
 ip access-group telnetting out
!
ip access-list extended telnetting
 remark Do not allow user1 subnet to telnet out
 deny tcp 172.20.0.0 0.0.255.255 any eq telnet
 remark Allow Top subnet to telnet out
 permit tcp 172.33.0.0 0.0.255.255 any eq telnet
```

Example: Filtering on TCP and ICMP Using Port Numbers

In the following example, the first line of the extended access list named acl1 permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 172.28.1.2. The last line permits incoming ICMP messages for error feedback.

```
interface gigabitethernet 0/0/0
 ip access-group acl1 in
!
ip access-list extended acl1
 permit tcp any 172.28.0.0 0.0.255.255 gt 1023
```

```
 permit tcp any host 172.28.1.2 eq 25
 permit icmp any 172.28.0.0 255.255.255.255
```

Example: Allowing SMTP E-mail and Established TCP Connections

Suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the gigabitethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the router always will accept mail connections on port 25 is what makes possible separate control of incoming and outgoing services. The access list can be configured on either the outbound or inbound interface.

In the following example, the gigabitethernet network is a Class B network with the address 172.18.0.0, and the address of the mail host is 172.18.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
interface gigabitethernet 0/0/0
 ip access-group 102 in
!
access-list 102 permit tcp any 172.18.0.0 0.0.255.255 established
access-list 102 permit tcp any host 172.18.1.2 eq 25
```

Example: Preventing Access to the Web by Filtering on Port Name

In the following example, the w1and w2 workstations are not allowed web access; other hosts on network 172.20.0.0 are allowed web access:

```
interface gigabitethernet0/0/0
 ip access-group no-web out
!
ip access-list extended no-web
 remark Do not allow w1 to browse the web
 deny host 172.20.3.85 any eq http
 remark Do not allow w2 to browse the web
 deny host 172.20.3.13 any eq http
 remark Allow others on our network to browse the web
 permit 172.20.0.0 0.0.255.255 any eq http
```

Example: Filtering on Source Address and Logging the Packets

The following example defines access lists 1 and 2, both of which have logging enabled:

```
interface gigabitethernet 0/0/0
 ip address 172.16.1.1 255.0.0.0
 ip access-group 1 in

!
access-list 1 permit 172.25.0.0 0.0.255.255 log
access-list 1 deny 172.30.0.0 0.0.255.255 log
!
```

```
access-list 2 permit 172.27.3.4 log
access-list 2 deny 172.17.0.0 0.0.255.255 log
```

If the interface receives 10 packets from 172.25.7.7 and 14 packets from 172.17.23.21, the first log will look like the following:

```
list 1 permit 172.25.7.7 1 packet
list 2 deny 172.17.23.21 1 packet
```

Five minutes later, the console will receive the following log:

```
list 1 permit 172.25.7.7 9 packets
list 2 deny 172.17.23.21 13 packets
```

Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44

Device# debug mpls ldp advertisements peer-acl acl1

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

Additional References Creating an IP Access List and Applying It to an Interface

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Related Topic	Document Title
• Order of access list entries • Access list entries based on time of day or week • Packets with noninitial fragments	Refining an IP Access List
Filtering on IP options, TCP flags, or noncontiguous ports	Creating an IP Access List for Filtering
Controlling logging-related parameters	Understanding Access Control List Loggingl

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards or RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

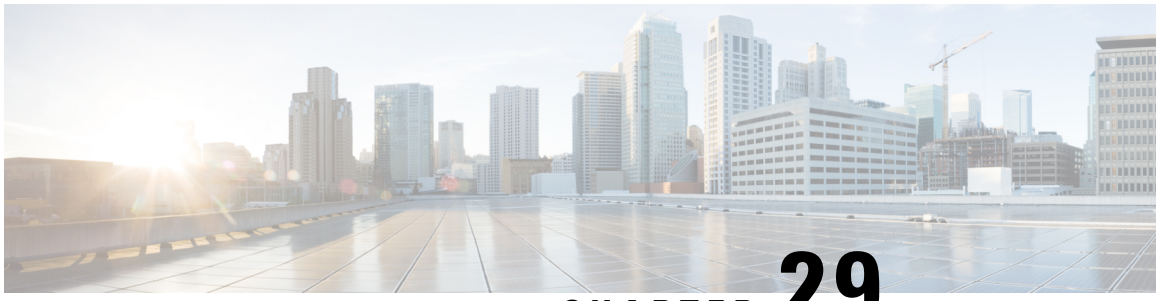
Feature Information for Creating an IP Access List and Applying It to an Interface

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 49: Feature Information for Creating IP Access Lists and Applying It to an Interface

Feature Name	Releases	Feature Configuration Information
ACL—Access Control List Source and Destination Address Matching	Cisco IOS XE Release 3.5S	In the Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
ACL—ICMP Code	Cisco IOS XE Release 3.5S	In the Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
ACL Performance Enhancement	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified for this feature.



CHAPTER 29

Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

This module describes how to use an IP access list to filter IP packets that contain certain IP Options, TCP flags, noncontiguous ports.

Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

Before you perform any of the tasks in this module, you should be familiar with the information in the following modules:

- "IP Access List Overview"
- "Creating an IP Access List and Applying It to an Interface"

Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

IP Options

IP uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Options, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for the most common communications. IP Options include provisions for time stamps, security, and special routing.

IP Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. IP Options can have one of two formats:

- Format 1: A single octet of option-type.
- Format 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet, the option-length octet, and the option-data octets.

The option-type octet is viewed as having three fields: a 1-bit copied flag, a 2-bit option class, and a 5-bit option number. These fields form an 8-bit value for the option type field. IP Options are commonly referred to by their 8-bit value.

For a complete list and description of IP Options, refer to RFC 791, *Internet Protocol* at the following URL: http://www.faqs.org/rfcs/rfc791.html

Benefits of Filtering IP Options

- Filtering of packets that contain IP Options from the network relieves downstream devices and hosts of the load from options packets.
- This feature also minimizes load to the Route Processor (RP) for packets with IP Options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Filtering the packets prevents them from impacting the RP.

Benefits of Filtering on TCP Flags

The ACL TCP Flags Filtering feature provides a flexible mechanism for filtering on TCP flags. Previously, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.

Because TCP packets can be sent as false synchronization packets that can be accepted by a listening port, it is recommended that administrators of firewall devices set up some filtering rules to drop false TCP packets.

The ACEs that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have a very specific group of TCP flags set or not set. The ACL TCP Flags Filtering feature provides a greater degree of packet-filtering control in the following ways:

- You can select any desired combination of TCP flags on which to filter TCP packets.
- You can configure ACEs to allow matching on a flag that is set, as well as on a flag that is not set.

TCP Flags

The table below lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

Table 50: TCP Flags

TCP Flag	Purpose
ACK	Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
FIN	Finish flag—Used to clear connections.
PSH	Push flag—Indicates the data in the call should be immediately pushed through to the receiving user.
RST	Reset flag—Indicates that the receiver should delete the connection without further interaction.
SYN	Synchronize flag—Used to establish connections.
URG	Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.

Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

This feature greatly reduces the number of access control entries (ACEs) required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, use this feature to consolidate existing groups of access list entries wherever it is possible and when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

How Filtering on TTL Value Works

IP extended named and numbered access lists may filter on the TTL value of packets arriving at or leaving an interface. Packets with any possible TTL values 0 through 255 may be permitted or denied (filtered). Like filtering on other fields, such as source or destination address, the **ip access-group** command specifies **in** or **out**, which makes the access list ingress or egress and applies it to incoming or outgoing packets, respectively. The TTL value is checked in conjunction with the specified protocol, application, and any other settings in the access list entry, and all conditions must be met.

Special Handling for Packets with TTL Value of 0 or 1 Arriving at an Ingress Interface

The software switching paths—distributed Cisco Express Forwarding (dCEF), CEF, fast switching, and process switching—will usually permit or discard the packets based on the access list statements. However, when the TTL value of packets arriving at an ingress interface have a TTL of 0 or 1, special handling is required. The packets with a TTL value of 0 or 1 get sent to the process level before the ingress access list is

checked in CEF, dCEF, or the fast switching paths. The ingress access list is applied to packets with TTL values 2 through 255 and a permit or deny decision is made.

Packets with a TTL value of 0 or 1 are sent to the process level because they will never be forwarded out of the device; the process level must check whether each packet is destined for the device and whether an Internet Control Message Protocol (ICMP) TTL Expire message needs to be sent back. This means that even if an ACL with TTL value 0 or 1 filtering is configured on the ingress interface with the intention to drop packets with a TTL of 0 or 1, the dropping of the packets will not happen in the faster paths. It will instead happen in the process level when the process applies the ACL. This is also true for hardware switching platforms. Packets with TTL value of 0 or 1 are sent to the process level of the route processor (RP) or Multilayer Switch Feature Card (MSFC).

On egress interfaces, access list filtering on TTL value works just like other access list features. The check will happen in the fastest switching path enabled in the device. This is because the faster switching paths handle all the TTL values (0 through 255) equally on the egress interface.

Control Plane Policing for Filtering TTL Values 0 and 1

The special behavior for packets with a TTL value of 0 or 1 results in higher CPU usage for the device. If you are filtering on TTL value of 0 or 1, you should use control plane policing (CPP) to protect the CPU from being overwhelmed. In order to leverage CPP, you must configure an access list especially for filtering TTL values 0 and 1 and apply the access list through CPP. This access list will be a separate access list from any other interface access lists. Because CPP works for the entire system, not just on individual interfaces, you would need to configure only one such special access list for the entire device. This task is described in the section "Enabling Control Plane Policing to Filter on TTL Values 0 and 1".

Benefits of Filtering on TTL Value

- Filtering on time-to-live (TTL) value provides a way to control which packets are allowed to reach the device or are prevented from reaching the device. By looking at your network layout, you can choose whether to accept or deny packets from a certain device based on how many hops away it is. For example, in a small network, you can deny packets from a location more than three hops away. Filtering on TTL value allows you to validate if the traffic originated from a neighboring device. You can accept only packets that reach you in one hop, for example, by accepting only packets with a TTL value of one less than the initial TTL value of a particular protocol.
- Many control plane protocols communicate only with their neighbors, but receive packets from everyone. By applying an access list that filters on TTL to receiving routers, you can block unwanted packets.
- The Cisco software sends all packets with a TTL value of 0 or 1 to the process level. The device must then send an Internet Control Message Protocol (ICMP) TTL value expire message to the source. By filtering packets that have a TTL value of 0 through 2, you can reduce the load on the process level.

How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

Filtering Packets That Contain IP Options

Complete these steps to configure an access list to filter packets that contain IP options and to verify that the access list has been configured correctly.

Note

- The ACL Support for Filtering IP Options feature can be used only with named, extended ACLs.
- Resource Reservation Protocol (RSVP) Multiprotocol Label Switching Traffic Engineering (MPLS TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP options packets may not function in drop or ignore mode if this feature is configured.
- On most Cisco devices, a packet with IP options is not switched in hardware, but requires control plane software processing (primarily because there is a need to process the options and rewrite the IP header), so all IP packets with IP options will be filtered and switched in software.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip access-list extended** *access-list-name*

Example:

```
Device(config)# ip access-list extended mylist1
```

Specifies the IP access list by name and enters named access list configuration mode.

Step 4 [*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# deny ip any any option traceroute
```

(Optional) Specifies a **deny** statement in named IP access list mode.

- This access list happens to use a **deny**statement first, but a **permit** statement could appear first, depending on the order of statements you need.

- Use the **option** keyword and *option-value* argument to filter packets that contain a particular IP Option.
- In this example, any packet that contains the traceroute IP option will be filtered out.
- Use the **no** *sequence-number* form of this command to delete an entry.

Step 5 [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# permit ip any any option security
```

Specifies a **permit** statement in named IP access list mode.

- In this example, any packet (not already filtered) that contains the security IP option will be permitted.
- Use the **no** *sequence-number* form of this command to delete an entry.

Step 6 Repeat Step 4 or Step 5 as necessary.

Allows you to revise the access list.

Step 7 **end**

Example:

```
Device(config-ext-nacl)# end
```

(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.

Step 8 **show ip access-lists** *access-list-name*

Example:

```
Device# show ip access-lists mylist1
```

(Optional) Displays the contents of the IP access list.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Note To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

Filtering Packets That Contain TCP Flags

This task configures an access list to filter packets that contain TCP flags and verifies that the access list has been configured correctly.

Note

- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco ACLs.
- Previously, the following command-line interface (CLI) format could be used to configure a TCP flag-checking mechanism:

permit tcp any any rst The following format that represents the same ACE can now be used: **permit tcp any any match-any +rst** Both the CLI formats are accepted; however, if the new keywords **match-all** or **match-any** are chosen, they must be followed by the new flags that are prefixed with "+" or "-". It is advisable to use only the old format or the new format in a single ACL. You cannot mix and match the old and new CLI formats.

Caution If a device having ACEs with the new syntax format is reloaded with a previous version of the Cisco software that does not support the ACL TCP Flags Filtering feature, the ACEs will not be applied, leading to possible security loopholes.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip access-list extended** *access-list-name*

Example:

```
Device(config)# ip access-list extended kmd1
```

Specifies the IP access list by name and enters named access list configuration mode.

Step 4 [*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established**|{**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# permit tcp any any match-any +rst
```

Specifies a **permit** statement in named IP access list mode.

- This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need.
- Use the TCP command syntax of the **permit**command.
- Any packet with the RST TCP header flag set will be matched and allowed to pass the named access list kmd1 in Step 3.

Step 5 [*sequence-number*] **deny tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established**|{**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# deny tcp any any match-all -ack -fin
```

(Optional) Specifies a **deny** statement in named IP access list mode.

- This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need.
- Use the TCP command syntax of the **deny**command.
- Any packet that does not have the ACK flag set, and also does not have the FIN flag set, will not be allowed to pass the named access list kmd1 in Step 3.
- See the **deny**(IP) command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP).

Step 6 Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number*command to delete an entry.

Allows you to revise the access list.

Step 7 **end**

Example:

```
Device(config-ext-nacl)# end
```

(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Step 8 **show ip access-lists** *access-list-name*

Example:

```
Device# show ip access-lists kmd1
```

(Optional) Displays the contents of the IP access list.

- Review the output to confirm that the access list includes the new entry.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Configuring an Access Control Entry with Noncontiguous Ports

Perform this task to create access list entries that use noncontiguous TCP or UDP port numbers. Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

Note The ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry feature can be used only with named, extended ACLs.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip access-list extended** *access-list-name*

Example:

```
Device(config)# ip access-list extended acl-extd-1
```

Specifies the IP access list by name and enters named access list configuration mode.

Step 4 [*sequence-number*] **permit tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679
```

Specifies a **permit** statement in named IP access list configuration mode.

- Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
- If the operator is positioned after the source and source-wildcard arguments, it must match the source port. If the operator is positioned after the destination and destination-wildcard arguments, it must match the destination port.
- The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq**operators. All other operators require one port number.
- To filter UDP ports, use the UDP syntax of this command.

Step 5 [*sequence-number*] **deny tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# deny tcp any neq 45 565 632 any
```

(Optional) Specifies a **deny** statement in named access list configuration mode.

- Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
- If the *operator* is positioned after the *source* and *source-wildcard* arguments, it must match the source port. If the *operator* is positioned after the *destination* and *destination-wildcard* arguments, it must match the destination port.
- The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq**operators. All other operators require one port number.
- To filter UDP ports, use the UDP syntax of this command.

Step 6 Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.

Allows you to revise the access list.

Step 7 **end**

Example:

```
Device(config-ext-nacl)# end
```

(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.

Step 8 **show ip access-lists** *access-list-name*

Example:

```
Device# show ip access-lists kmd1
```

(Optional) Displays the contents of the access list.

Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry

Perform this task to consolidate a group of access list entries with noncontiguous ports into one access list entry.

Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip access-lists** *access-list-name*

Example:

```
Device# show ip access-lists mylist1
```

(Optional) Displays the contents of the IP access list.

- Review the output to see if you can consolidate any access list entries.

Step 3 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 4 **ip access-list extended** *access-list-name*

Example:

```
Device(config)# ip access-list extended mylist1
```

Specifies the IP access list by name and enters named access list configuration mode.

Step 5 **no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard*[**option** *option-name*] [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# no 10
```

Removes the redundant access list entry that can be consolidated.

- Repeat this step to remove entries to be consolidated because only the port numbers differ.
- After this step is repeated to remove the access list entries 20, 30, and 40, for example, those entries are removed because they will be consolidated into one **permit** statement.
- If a *sequence-number* is specified, the rest of the command syntax is optional.

Step 6 [*sequence-number*] **permit** *protocol source source-wildcard*[*operator port*[*port*]] *destination destination-wildcard*[*operator port*[*port*]] [**option** *option-name*] [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Example:

```
Device(config-ext-nacl)# permit tcp any neq 45 565 632 any eq 23 45 34 43
```

Specifies a **permit** statement in named access list configuration mode.

- In this instance, a group of access list entries with noncontiguous ports was consolidated into one **permit** statement.
- You can configure up to 10 ports after the **eq** and **neq** operators.

Step 7 Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no** *sequence-number* command to delete an entry.

Allows you to revise the access list.

Step 8 **end**

Example:

```
Device(config-std-nacl)# end
```

(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.

Step 9 **show ip access-lists** *access-list-name*

Example:

```
Device# show ip access-lists mylist1
```

(Optional) Displays the contents of the access list.

What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Filtering Packets Based on TTL Value

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

Note When the access list specifies the operation EQ or NEQ, depending on the Cisco software release in use on the device, the access lists can specify up to ten TTL values. The number of TTL values can vary by the Cisco software release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard*[**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **interface** *type number*
8. **ip access-group** *access-list-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard*[**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement. • This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	--
Step 6	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the command-line interface (CLI) mode hierarchy.
Step 7	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Configures an interface type and enters interface configuration mode.
Step 8	**ip access-group** *access-list-name* {**in** | **out**} **Example:** `Device(config-if)# ip access-group ttlfilter in`	Applies the access list to an interface.

Enabling Control Plane Policing to Filter on TTL Values 0 and 1

Perform this task to filter IP packets based on a TTL value of 0 or 1 and to protect the CPU from being overwhelmed. This task configures an access list for classification on TTL value 0 and 1, configures the Modular QoS Command-Line Interface (CLI) (MQC), and applies a policy map to the control plane. Any packets that pass the access list are dropped. This special access list is separate from any other interface access lists.

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* **ttl** *operator* value
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **class-map** *class-map-name* [**match-all** | **match-any**]
8. **match access-group** {*access-group* | **name** *access-group-name*}
9. **exit**
10. **policy-map** *policy-map-name*
11. **class** {*class-name* | **class-default**}
12. **drop**
13. **exit**
14. **exit**
15. **control-plane**
16. **service-policy** {**input** | **output**} *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on a TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* **ttl** *operator* value **Example:**	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement.

	Command or Action	Purpose
	`Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	• This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	The packets that pass the access list will be dropped.
Step 6	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 7	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Device(config)# class-map acl-filtering`	Creates a class map to be used for matching packets to a specified class.
Step 8	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Device(config-cmap)# match access-group name ttlfilter`	Configures the match criteria for a class map on the basis of the specified access control list.
Step 9	**exit** **Example:** `Device(config-cmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 10	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map acl-filter`	Creates or modifies a policy map that can be attached to one or more interface to specify a service policy.
Step 11	**class** {*class-name* | **class-default**} **Example:** `Device(config-pmap)# class acl-filter-class`	Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the class-default class) before you configure its policy.
Step 12	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 13	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 14	**exit** **Example:** `Device(config-pmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 15	**control-plane** **Example:** `Device(config)# control-plane`	Associates or modifies attributes or parameters that are associated with the control plane of the device.
Step 16	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Device(config-cp)# service-policy input acl-filter`	Attaches a policy map to a control plane for aggregate control plane services.

Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports

Example: Filtering Packets That Contain IP Options

The following example shows an extended access list named mylist2 that contains access list entries (ACEs) that are configured to permit TCP packets only if they contain the IP Options that are specified in the ACEs:

```
ip access-list extended mylist2
 10 permit ip any any option eool
 20 permit ip any any option record-route
 30 permit ip any any option zsu
 40 permit ip any any option mtup
```

The **show access-list** command has been entered to show how many packets were matched and therefore permitted:

```
Device# show ip access-list mylist2
Extended IP access list test
10 permit ip any any option eool (1 match)
20 permit ip any any option record-route (1 match)
30 permit ip any any option zsu (1 match)
40 permit ip any any option mtup (1 match)
```

Example: Filtering Packets That Contain TCP Flags

The following access list allows TCP packets only if the TCP flags ACK and SYN are set and the FIN flag is not set:

```
ip access-list extended aaa
```

```
 permit tcp any any match-all +ack +syn -fin
 end
```

The **show access-list** command has been entered to display the ACL:

```
Device# show access-list aaa

Extended IP access list aaa
 10 permit tcp any any match-all +ack +syn -fin
```

Example: Creating an Access List Entry with Noncontiguous Ports

The following access list entry can be created because up to ten ports can be entered after the **eq** and **neq** operators:

```
ip access-list extended aaa
 permit tcp any eq telnet ftp any eq 23 45 34
 end
```

Enter the **show access-lists** command to display the newly created access list entry.

```
Device# show access-lists aaa

Extended IP access list aaa
 10 permit tcp any eq telnet ftp any eq 23 45 34
```

Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports

The **show access-lists** command is used to display a group of access list entries for the access list named abc:

```
Device# show access-lists abc
Extended IP access list abc
 10 permit tcp any eq telnet any eq 450
 20 permit tcp any eq telnet any eq 679
 30 permit tcp any eq ftp any eq 450
 40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended abc
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
 end
```

When the **show access-lists** command is reentered, the consolidated access list entry is displayed:

```
Device# show access-lists abc
```

```
Extended IP access list abc
 10 permit tcp any eq telnet ftp any eq 450 679
```

Example: Filtering on TTL Value

The following access list filters IP packets containing type of service (ToS) level 3 with time-to-live (TTL) values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL value not equal to 1, and it sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended incomingfilter
 deny ip any any tos 3 ttl eq 10 20
 deny ip any any ttl gt 154 fragments
 permit ip any any precedence flash ttl neq 1 log
!
interface ethernet 0

ip access-group incomingfilter in
```

Example: Control Plane Policing to Filter on TTL Values 0 and 1

The following example configures a traffic class called acl-filter-class for use in a policy map called acl-filter. An access list permits IP packets from any source having a time-to-live (TTL) value of 0 or 1. Any packets matching the access list are dropped. The policy map is attached to the control plane.

```
ip access-list extended ttlfilter

 permit ip any any ttl eq 0 1

class-map acl-filter-class

 match access-group name ttlfilter

policy-map acl-filter

 class acl-filter-class

 drop

control-plane

 service-policy input acl-filter
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*
Configuring the device to drop or ignore packets containing IP Options by using the **no ip options** command.	*ACL IP Options Selective Drop*
Overview information about access lists.	*IP Access List Overview*
Information about creating an IP access list and applying it to an interface	*Creating an IP Access List and Applying It to an Interface*
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*

RFCs

RFC	Title
RFC 791	*Internet Protocol* http://www.faqs.org/rfcs/rfc791.html
RFC 793	*Transmission Control Protocol*
RFC 1393	*Traceroute Using an IP Option*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

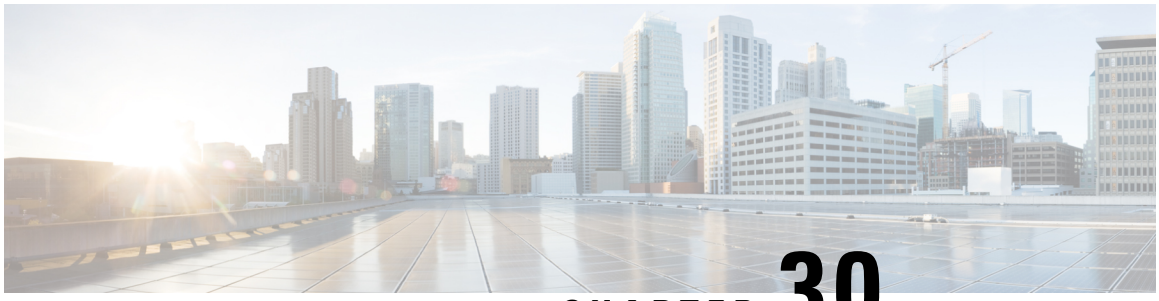
Feature Information for Creating an IP Access List to Filter

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 51: Feature Information for Creating an IP Access List to Filter

Feature Name	Releases	Feature Configuration Information
ACL--Named ACL Support for Noncontiguous Ports on an Access Control Entry	12.3(7)T 12.2(25)S	This feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports.
ACL Support for Filtering IP Options	12.3(4)T 12.2(25)S 15.2(2)S 15.4(1)S	This feature allows you to filter packets having IP Options, in order to prevent routers from becoming saturated with spurious packets. In Cisco IOS Release 15.4(1)S, support was added for the Cisco ASR 901S series routers.
ACL TCP Flags Filtering	12.3(4)T 12.2(25)S	This feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.



CHAPTER 30

Configuring an FQDN ACL

This document describes how to configure an access control lists (ACL) using a fully qualified domain name (FQDN). The Configuring an FQDN ACL feature allows you to configure and apply an ACL to a wireless session based on the domain name system (DNS). The domain names are resolved to IP addresses, the IP addresses are given to the client as part of the DNS response, and the FQDN is then mapped to an ACL based on the IP address.

Restrictions for Configuring FQDN ACL

The Configuring FQDN ACL feature is supported only on IPv4 wireless sessions.

Information About Configuring an FQDN ACL

Configuring an FQDN ACL

When access control lists (ACLs) are configured using a fully qualified domain name (FQDN), ACLs can be applied based on the destination domain name. The destination domain name is then resolved to an IP address, which is provided to the client as a part of the DNS response.

Guest users can log in using web authentication with a parameter map that consists of an FQDN ACL name.

Before you configure an FQDN ACL, complete the following tasks:

- Configure an IP access list.
- Configure an IP domain name list.
- Map an FQDN ACL with a domain name.

You can apply an access list to a specific domain by configuring the RADIUS server to send the **fqdn-acl-name** AAA attribute to the controller. The operating system checks for the passthrough domain list and its mapping, and permits the FQDN. The FQDN ACL allows clients to access only configured domains without authentication.

Note By default, an IP access list name is configured with the same name as the pass-through domain name. To override the default name, you can use the **access-session passthrou-access-group** *access-group-name* **passthrou-domain-list** *domain-list-name* command in global configuration mode.

How to Configure FQDN ACL

Configuring an IP Access List

SUMMARY STEPS

1. **ip access-list extended** *name*
2. **permit ip any any**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip access-list extended** *name* **Example:** `Device (config)# ip access-list extended ABC`	Creates the IP access list.
Step 2	**permit ip any any** **Example:** `Device (config-ext-nacl)# permit ip any any`	Specifies the domains to be allowed for the wireless client. The domains are specified in the domain name list.

Configuring a Domain Name List

You can configure a domain name list that contains a list of domain names that are allowed for DNS snooping by the access point. The DNS domain list name string must be identical to the extended access list name.

SUMMARY STEPS

1. **passthrou-domain-list** *name*
2. **match** *word*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**passthrou-domain-list** *name*
	Example:
	`Device (config)# passthrou-domain-list abc`
	`Device (config-fqdn-acl-domains)#`
		Configures a passthrough domain name list.

Step 2	**match** *word*
	Example:
	`Device (config-fqdn-acl-domains)# match play.google.com`
	`Device (config-fqdn-acl-domains)# match www.yahoo.com`
		Configures a passthrough domain list. Adds a list of websites that the client is allowed to query for access without first being required to be authenticated through the RADIUS server.

Mapping the FQDN ACL with a Domain Name

SUMMARY STEPS

1. **access-session passthrou-access-group** *access-group-name* **passthrou-domain-list** *domain-list-name*
2. **parameter-map type webauth** *domain-list-name* and **login-auth-bypass fqdn-acl-name** *acl-name* **domain-name** *domain-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**access-session passthrou-access-group** *access-group-name* **passthrou-domain-list** *domain-list-name* **Example:** `Device (config)# access-session passthrou-access-group abc passthrou-domain-list abc`	Maps the FQDN ACL AAA attribute name with the domain name list. Use this command when configuring central web authentication.
Step 2	**parameter-map type webauth** *domain-list-name* and **login-auth-bypass fqdn-acl-name** *acl-name* **domain-name** *domain-name* **Example:** `Device (config)# parameter-map type webauth abc SwitchControllerDevice (config-params-parameter-map)# login-auth-bypass fqdn-acl-name abc domain-name abc`	Maps an FQDN ACL name with the domain name list. Use the command when configuring local authentication on the controller. The RADIUS server can be configured to return an FQDN ACL name as part of the authenticated user profile. The controller dynamically applies the FQDN ACL to the user if the FQDN ACL is defined on the controller.

Monitoring an FQDN ACL

The following commands can be used to monitor FQDN ACLs.

Command	Purpose
show access-session interface *interface-name* **details**	Displays the FQDN ACL information configured on the interface.
show access-session fqdn fqdn-maps	Displays the FQDN ACL mapped to the domain name list.
show access-session fqdn list-domain *domain-name*	Displays the domain names.
show access-session fqdn passthru-domain-list	Displays the domains that are configured.

Configuration Examples for an FQDN ACL

Examples: FQDN ACL Configuration

This example shows how to create IP access list:

```
# config terminal
(config)# ip access-list extended abc
(config-ext-nacl)# permit ip any any
(config-ext-nacl)# end
# show ip access-list abc
```

This example shows how to configure domain name list:

```
# config terminal
(config)# passthrou-domain-list abc
(config-fqdn-acl-domains)# match play.google.com
(config-fqdn-acl-domains)# end
# show access-session fqdn fqdn-maps
```

This example shows how to map FQDN ACL with domain name using central web authentication:

```
# config terminal
(config)# access-session passthrou-access-group abc passthrou-domain-list abc
(config)# end
# show access-session interface vlan 20
```

This example shows how to map FQDN ACL with domain name using local authentication:

```
# config terminal
(config)# parameter-map type webauth abc
(config-params-parameter-map)# login-auth-bypass fqdn-acl-name abc domain-name abc
(config-params-parameter-map)# end
# show access-session fqdn fqdn-maps
```

Additional References for Configuring FQDN ACL

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
ACL configuration guide	*Security Configuration Guide: Access Control Lists*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

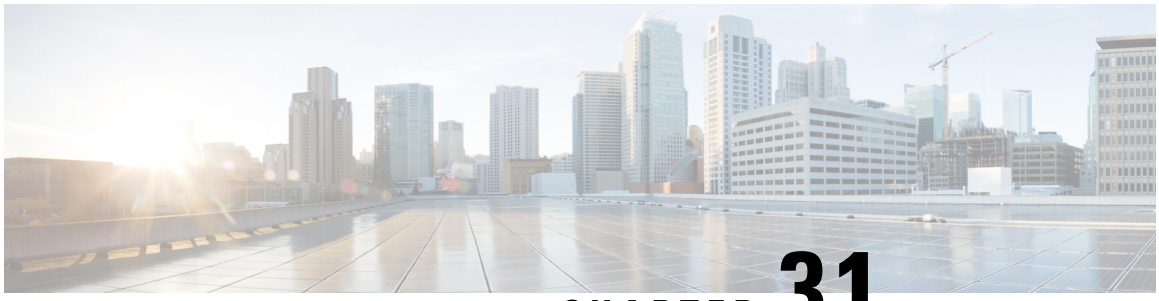
Feature Information for Configuring FQDN ACL

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 52: Feature Information for Configuring FQDN ACL

Feature Name	Releases	Feature Information
Configuring an FQDN ACL		The Configuring an FQDN ACL feature allows you to configure and apply an access control lists (ACL) to a wireless session based on the domain name system (DNS). The domain names are resolved to IP addresses, where the IP addresses are given to the client as part of the DNS response; the FQDN is then mapped to an ACL based on the IP address. The following commands were introduced or modified: **access session passthrou access group**, **login-auth-bypass**, **parameter-map type webauth global**, **pass throu domain list name**, **show access-session fqdn**.



CHAPTER 31

Refining an IP Access List

There are several ways to refine an access list while or after you create it. You can change the order of the entries in an access list or add entries to an access list. You can restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering noninitial fragments of packets.

Information About Refining an IP Access List

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

Sequence numbers allow users to add access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Benefits of Access List Sequence Numbers

An access list sequence number is a number at the beginning of a **permit** or **deny** command in an access list. The sequence number determines the order that the entry appears in the access list. The ability to apply sequence numbers to IP access list entries simplifies access list changes.

Prior to having sequence numbers, users could only add access list entries to the end of an access list; therefore, needing to add statements anywhere except the end of the list required reconfiguring the entire access list. There was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed,

then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry. Sequence numbers make revising an access list much easier.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- This feature works with named and numbered, standard and extended IP access lists.

Benefits of Time Ranges

Benefits and possible uses of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including the following:
  - Perimeter security using access lists
  - Data confidentiality with IP Security Protocol (IPsec)
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.

- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Benefits Filtering Noninitial Fragments of Packets

Filter noninitial fragments of packets with an extended access list if you want to block more of the traffic you intended to block, not just the initial fragment of such packets. You should first understand the following concepts.

If the **fragments**keyword is used in additional IP access list entries that deny fragments, the fragment control feature provides the following benefits:

Additional Security

You are able to block more of the traffic you intended to block, not just the initial fragment of such packets. The unwanted fragments no longer linger at the receiver until the reassembly timeout is reached because they are blocked before being sent to the receiver. Blocking a greater portion of unwanted traffic improves security and reduces the risk from potential hackers.

Reduced Cost

By blocking unwanted noninitial fragments of packets, you are not paying for traffic you intended to block.

Reduced Storage

By blocking unwanted noninitial fragments of packets from ever reaching the receiver, that destination does not have to store the fragments until the reassembly timeout period is reached.

Expected Behavior Is Achieved

The noninitial fragments will be handled in the same way as the initial fragment, which is what you would expect. There are fewer unexpected policy routing results and fewer fragments of packets being routed when they should not be.

Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has...	Then...

...no **fragments** keyword (the default), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information:
	• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.
	For an access list entry that contains Layer 3 and Layer 4 information:
	• The entry is applied to nonfragmented packets and initial fragments.
	  • If the entry is a **permit** statement, then the packet or fragment is permitted.
	  • If the entry is a **deny** statement, then the packet or fragment is denied.
	• The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and
	  • If the entry is a **permit** statement, then the noninitial fragment is permitted.
	  • If the entry is a **deny** statement, then the next access list entry is processed.
	Note The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.

...the **fragments** keyword, and assuming all of the access-list entry information matches,	The access list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access list entry that contains any Layer 4 information.

Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

How to Refine an IP Access List

The tasks in this module provide you with various ways to refine an access list if you did not already do so while you were creating it. You can change the order of the entries in an access list, add entries to an access

list, restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

Revising an Access List Using Sequence Numbers

Perform this task if you want to add entries to an existing access list, change the order of entries, or simply number the entries in an access list to accommodate future changes.

Note Remember that if you want to delete an entry from an access list, you can simply use the **no deny** or **no permit** form of the command, or the **no** *sequence-number* command if the statement already has a sequence number.

Note
- Access list sequence numbers do not support dynamic, reflexive, or firewall access lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list** {**standard**| **extended**} *access-list-name*
5. Do one of the following:
   - *sequence-number* **permit** *source source-wildcard*
   - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Do one of the following:
   - *sequence-number* **deny** *source source-wildcard*
   - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:** `Router(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers. • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
Step 4	**ip access-list** {**standard**| **extended**} *access-list-name* **Example:** `Router(config)# ip access-list standard xyz123`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you specify subsequent **permit** and **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you specify subsequent **permit** and **deny** statements using the extended access list syntax.
Step 5	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0.255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **permit** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended **permit** command syntax.
Step 6	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:**	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **deny** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP).

	Command or Action	Purpose
	`Router(config-std-nacl)# 110 deny 10.6.6.7 0.0.0.255`	• Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended **deny** command syntax.
Step 7	Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 9	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists xyz123`	(Optional) Displays the contents of the IP access list. • Review the output to see that the access list includes the new entry.

Examples

The following is sample output from the **show ip access-lists** command when the **xyz123** access list is specified.

```
Router# show ip access-lists xyz123
Standard IP access list xyz123
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.5, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Restricting an Access List Entry to a Time of Day or Week

By default, access list statements are always in effect once they are applied. However, you can define the times of the day or week that **permit** or **deny** statements are in effect by defining a time range, and then referencing the time range by name in an individual access list statement. IP and Internetwork Packet Exchange (IPX) named or numbered extended access lists can use time ranges.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*

4. [*sequence-number*] **deny** *protocol source*[*source-wildcard*] [*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]]
5. [*sequence-number*] **deny** *protocol source*[*source-wildcard*][*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]] **fragments**
6. [*sequence-number*] **permit** *protocol source*[*source-wildcard*] [*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]]
7. Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.
8. **end**
9. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended rstrct4`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	[*sequence-number*] **deny** *protocol source*[*source-wildcard*] [*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]] **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • This statement will apply to nonfragmented packets and initial fragments.
Step 5	[*sequence-number*] **deny** *protocol source*[*source-wildcard*][*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]] **fragments** **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1 fragments`	(Optional) Denies any packet that matches all of the conditions specified in the statement • This statement will apply to noninitial fragments.
Step 6	[*sequence-number*] **permit** *protocol source*[*source-wildcard*] [*operator port*[*port*]] *destination*[*destination-wildcard*] [*operator port*[*port*]]	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement.

	Command or Action	Purpose
	Example: `Router(config-ext-nacl)# permit tcp any any`	• If the *source-wildcard* or *destination-wildcard*isomitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard*or *destination destination-wildcard*to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Router(config-ext-nacl)# end`	Ends configuration mode and returns the system to privileged EXEC mode.
Step 9	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Note To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

Configuration Examples for Refining an IP Access List

Example Resequencing Entries in an Access List

The following example shows an access list before and after resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list carls
Extended IP access list carls
    10 permit ip host 10.3.3.3 host 172.16.5.34
```

```
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any
Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
Router# show access-list carls
Extended IP access list carls
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Example Adding an Entry with a Sequence Number

In the following example, an new entry (sequence number 15) is added to an access list:

```
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
Router(config)# ip access-list standard tryon
Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Example Adding an Entry with No Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Router# show access-list
Standard IP access list resources
```

```
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255
```

Example Time Ranges Applied to IP Access List Entries

The following example creates a time range called no-http, which extends from Monday to Friday from 8:00 a.m. to 6:00 p.m. That time range is applied to the **deny** statement, thereby denying HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.

The time range called udp-yes defines weekends from noon to 8:00 p.m. That time range is applied to the **permit** statement, thereby allowing UDP traffic on Saturday and Sunday from noon to 8:00 p.m. only. The access list containing both statements is applied to inbound packets on Fast Ethernet interface 0/0/0.

```
time-range no-http
 periodic weekdays 8:00 to 18:00
!
time-range udp-yes
 periodic weekend 12:00 to 20:00
!
ip access-list extended strict
 deny tcp any any eq http time-range no-http
 permit udp any any time-range udp-yes
!
interface fastethernet 0/0/0
 ip access-group strict in
```

Example Filtering IP Packet Fragments

In the following access list, the first statement will deny only noninitial fragments destined for host 172.16.1.1. The second statement will permit only the remaining nonfragmented and initial fragments that are destined for host 172.16.1.1 TCP port 80. The third statement will deny all other traffic. In order to block noninitial fragments for any TCP port, we must block noninitial fragments for all TCP ports, including port 80 for host 172.16.1.1. That is, non-initial fragments will not contain Layer 4 port information, so, in order to block such traffic for a given port, we have to block fragments for all ports.

```
access-list 101 deny ip any host 172.16.1.1 fragments
access-list 101 permit tcp any host 172.16.1.1 eq 80
access-list 101 deny ip any any
```

Additional References

Related Documents

Related Topic	Document Title
Using the **time-range** command to establish time ranges	The chapter *Performing Basic System Management* in the *Cisco IOS XE Network Management Configuration Guide*
Network management command descriptions	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

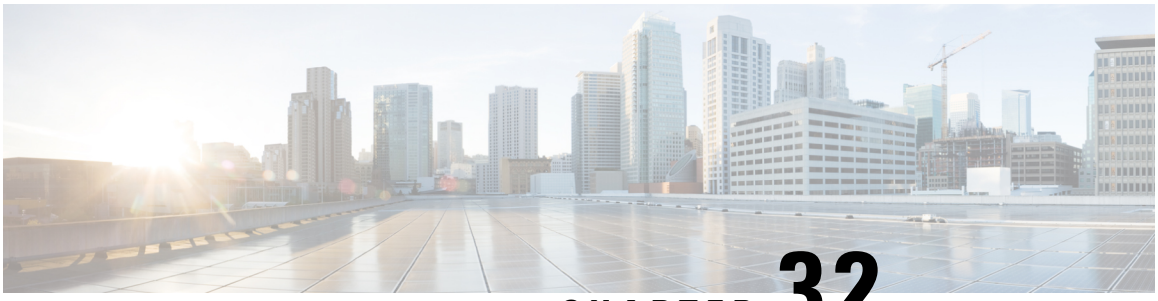
Feature Information for Refining an IP Access List

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 53: Feature Information for Refining an IP Access List

Feature Name	Releases	Feature Configuration Information
Time-Based Access Lists	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified for this feature.



CHAPTER 32

IP Named Access Control Lists

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

The IP Named Access Control Lists feature gives network administrators the option of using names to identify their access lists.

This module describes IP named access lists and how to configure them.

Information About IP Named Access Control Lists

Definition of an Access List

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets.

Access lists are identified and referenced by a name or a number. Access lists act as packet filters, filtering packets based on the criteria defined in each access list.

After you configure an access list, for the access list to take effect, you must either apply the access list to an interface (by using the **ip access-group** command), a vty (by using the **access-class** command), or reference

the access list by any command that accepts an access list. Multiple commands can reference the same access list.

In the following configuration, an IP access list named branchoffices is configured on Fast Ethernet interface 0/1/0 and applied to incoming packets. Networks other than the ones specified by the source address and mask pair cannot access Fast Ethernet interface 0/1/0. The destinations for packets coming from sources on network 172.16.7.0 are unrestricted. The destination for packets coming from sources on network 172.16.2.0 must be 172.31.5.4.

```
ip access-list extended branchoffices
 10 permit 172.16.7.0 0.0.0.3 any
 20 permit 172.16.2.0 0.0.0.255 host 172.31.5.4
!
interface fastethernet 0/1/0
 ip access-group branchoffices in
```

Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a task. You can reorder statements in or add statements to a named access list.

Named access lists support the following features that are not supported by numbered access lists:

- IP options filtering
- Noncontiguous ports
- TCP flag filtering
- Deleting of entries with the **no permit** or **no deny** command

Note Not all commands that accept a numbered access list will accept a named access list. For example, vty uses only numbered access lists.

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.

- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.
- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.
- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Access List Rules

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.

- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list**command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit**or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

- Before you add new ACL statements, provide time to the parser to clean up the deletion.

Where to Apply an Access List

You can apply access lists to the inbound or outbound interfaces of a device. Applying an access list to an inbound interface controls the traffic that enters the interface and applying an access list to an outbound interface controls the traffic that exits the interface.

When software receives a packet at the inbound interface, the software checks the packet against the statements that are configured for the access list. If the access list permits packets, the software processes the packet. Applying access lists to filter incoming packets can save device resources because filtered packets are discarded before entering the device.

Access lists on outbound interfaces filter packets that are transmitted (sent) out of the interface. You can use the TCP Access Control List (ACL) Splitting feature of the Rate-Based Satellite Control Protocol (RBSCP) on the outbound interface to control the type of packets that are subject to TCP acknowledgment (ACK) splitting on an outbound interface.

You can reference an access list by using a **debug** command to limit the amount of debug logs. For example, based on the filtering or matching criteria of the access list, debug logs can be limited to source or destination addresses or protocols.

You can use access lists to control routing updates, dial-on-demand (DDR), and quality of service (QoS) features.

How to Configure IP Named Access Control Lists

Creating an IP Named Access List

You can create an IP named access list to filter source addresses and destination addresses or a combination of addresses and other IP fields. Named access lists allow you to identify your access lists with an intuitive name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. **remark** *remark*
5. **deny** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**]
6. **remark** *remark*
7. **permit** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**]
8. Repeat Steps 4 through 7 to specify more statements for your access list.
9. **end**
10. **show ip access-lists**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended acl1`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark protect server by denying sales access to the acl1 network`	(Optional) Adds a description for an access list statement. • A remark can precede or follow an IP access list entry. • In this example, the **remark** command reminds the network administrator that the **deny** command configured in Step 5 denies the Sales network access to the interface.
Step 5	**deny** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**] **Example:** `Device(config-ext-nacl)# deny ip 192.0.2.0 0.0.255.255 host 192.0.2.10 log`	(Optional) Denies all packets that match all conditions specified by the remark.
Step 6	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark allow TCP from any source to any destination`	(Optional) Adds a description for an access list statement. • A remark can precede or follow an IP access list entry.
Step 7	**permit** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	Permits all packets that match all conditions specified by the statement.
Step 8	Repeat Steps 4 through 7 to specify more statements for your access list.	**Note** All source addresses that are not specifically permitted by a statement are denied by an implicit deny statement at the end of the access list.
Step 9	**end** **Example:**	Exits extended named access list configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-ext-nacl)# end`	
Step 10	**show ip access-lists** **Example:** `Device# show ip access-lists`	Displays the contents of all current IP access lists.

Example:

The following is sample output from the **show ip access-lists** command:

```
Device# show ip access-lists acl1

Extended IP access list acl1
   permit tcp any 192.0.2.0 255.255.255.255 eq telnet
   deny tcp any any
   deny udp any 192.0.2.0 255.255.255.255 lt 1024
   deny ip any any log
```

Applying an Access List to a Physical Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
5. **ip access-list extended** *acl-name acl-number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:**	Applies the specified access list to the inbound interface. • To filter source addresses, apply the access list to the inbound interface.

	Command or Action	Purpose
	`Device(config-if)# ip access-group acl1 in`	
Step 5	**ip access-list extended** *acl-name acl-number* **Example:**	Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list. Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list. • Access list to which all commands entered from ACL configuration mode apply, using an alphanumeric string of up to 30 characters, beginning with a letter. • Access list to which all commands entered from access list configuration mode apply, using a numeric identifier. For extended access lists, the valid range is 100 to 199.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IP Named Access Control Lists

Example: Creating an IP Named Access Control List

```
Device# configure terminal
Device(config)# ip access-list extended acl1
Device(config-ext-nacl)# remark protect server by denying sales access to the acl1 network
Device(config-ext-nacl)# deny ip 192.0.2.0 0.0.255.255 host 192.0.2.10 log
Device(config-ext-nacl)# remark allow TCP from any source to any destination
Device(config-ext-nacl)# permit tcp any any
```

Example: Applying the Access List to an Interface

```
Device# configure terminal

Device(config-if)# ip access-group acl1 in
```

Additional References for IP Named Access Control Lists

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

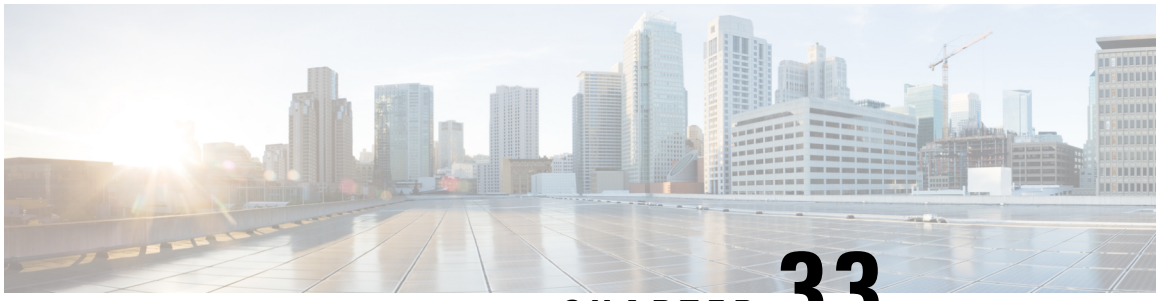
Feature Information for IP Named Access Control Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 54: Feature Information for IP Named Access Control Lists

Feature Name	Releases	Feature Information
IP Named Access Control Lists		Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.



CHAPTER 33

Commented IP Access List Entries

The Commented IP Access List Entries feature allows you to include comments or remarks about **deny** or **permit** conditions in any IP access list. These remarks make access lists easier for network administrators to understand. Each remark is limited to 100 characters in length.

This module provides information about the Commented IP Access List Entries feature.

../topics/Information About Commented IP Access List Entries

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.
- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.
- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access

lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.

- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
 remark Do not allow host1 subnet to telnet out
 deny tcp host 172.16.2.88 any eq telnet
```

How to Configure Commented IP Access List Entries

Writing Remarks in a Named or Numbered Access List

You can use a named or numbered access list configuration. You must apply the access list to an interface or terminal line after the access list is created for the configuration to work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} {*name* | *number*}
4. **remark** *remark*
5. **deny** *protocol* **host** *host-address* **any eq** *port*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} {*name* | *number*} **Example:** `Device(config)# ip access-list extended telnetting`	Identifies the access list by a name or number and enters extended named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out`	Adds a remark for an entry in a named IP access list. • The remark indicates the purpose of the **permit** or **deny** statement.
Step 5	**deny** *protocol* **host** *host-address* **any eq** *port* **Example:** `Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet`	Sets conditions in a named IP access list that denies packets.
Step 6	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended named access list configuration mode and enters privileged EXEC mode.

Configuration Examples for Commented IP Access List Entries

Example: Writing Remarks in an IP Access List

```
Device# configure terminal
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out
Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet
Device(config-ext-nacl)# end
```

Additional References for Commented IP Access List Entries

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

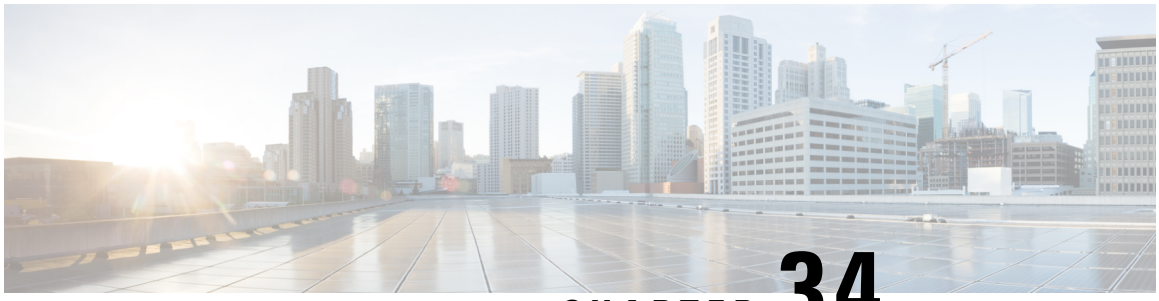
Feature Information for Commented IP Access List Entries

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 55: Feature Information for Commented IP Access List Entries

Feature Name	Releases	Feature Information
Commented IP Access List Entries		The Commented IP Access List Entries feature allows you to include comments or remarks about **deny** or **permit** conditions in any IP access list. These remarks make access lists easier for network administrators to understand. Each remark is limited to 100 characters in length. The following command was introduced or modified: **remark**.



CHAPTER 34

Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list logs an information message about the packet at the device console.

This module provides information about standard IP access list logging.

Restrictions for Standard IP Access List Logging

IP access list logging is supported only for routed interfaces or router access control lists (ACLs).

Information About Standard IP Access List Logging

Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list causes an information log message about the packet to be sent to the device console. The log level of messages that are printed to the device console is controlled by the **logging console** command.

The first packet that the access list inspects triggers the access list to log a message at the device console. Subsequent packets are collected over 5-minute intervals before they are displayed or logged. Log messages include information about the access list number, the source IP address of packets, the number of packets from the same source that were permitted or denied in the previous 5-minute interval, and whether a packet was permitted or denied. You can also monitor the number of packets that are permitted or denied by a particular access list, including the source address of each packet.

How to Configure Standard IP Access List Logging

Creating a Standard IP Access List Using Numbers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} **host** *address* [**log**]
4. **access-list** *access-list-number* {**deny** | **permit**} **any** [**log**]
5. **interface** *type number*
6. **ip access-group** *access-list-number* {**in** | **out**}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} **host** *address* [**log**] **Example:** `Device(config)# access-list 1 permit host 10.1.1.1 log`	Defines a standard numbered IP access list using a source address and wildcard, and configures the logging of informational messages about packets that match the access list entry at the device console.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} **any** [**log**] **Example:** `Device(config)# access-list 1 permit any log`	Defines a standard numbered IP access list by using an abbreviation for the source and source mask 0.0.0.0 255.255.255.255.
Step 5	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.
Step 6	**ip access-group** *access-list-number* {**in** | **out**} **Example:** `Device(config-if)# ip access-group 1 in`	Applies the specified numbered access list to the incoming or outgoing interface. • When you filter based on source addresses, you typically apply the access list to an incoming interface.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Creating a Standard IP Access List Using Names

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
4. {**deny** | **permit**} {**host** *address* | **any**} **log**
5. **exit**
6. **interface** *type number*
7. **ip access-group** *access-list-name* {**in** | **out**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list standard** *name* **Example:** `Device(config)# ip access-list standard acl1`	Defines a standard IP access list and enters standard named access list configuration mode.
Step 4	{**deny**	**permit**} {**host** *address*	**any**} **log** **Example:** `Device(config-std-nacl)# permit host 10.1.1.1 log`	Sets conditions in a named IP access list that will deny packets from entering a network or permit packets to enter a network, and configures the logging of informational messages about packets that match the access list entry at the device console.
Step 5	**exit** **Example:** `Device(config-std-nacl)# exit`	Exits standard named access list configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip access-group** *access-list-name* {**in** | **out**} **Example:** `Device(config-if)# ip access-group acl1 in`	Applies the specified access list to the incoming or outgoing interface. • When you filter based on source addresses, you typically apply the access list to an incoming interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Standard IP Access List Logging

Example: Creating a Standard IP Access List Using Numbers

```
Device# configure terminal
Device(config)# access-list 1 permit host 10.1.1.1 log
Device(config)# access-list 1 permit any log

Device(config-if)# ip access-group 1 in
```

Example: Creating a Standard IP Access List Using Names

```
Device# configure terminal
Device(config)# ip access-list standard acl1
Device(config-std-nacl)# permit host 10.1.1.1 log
Device(config-std-nacl)# exit

Device(config-if)# ip access-group acl1 in
```

Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44

Device# debug mpls ldp advertisements peer-acl acl1

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

Additional References for Standard IP Access List Logging

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

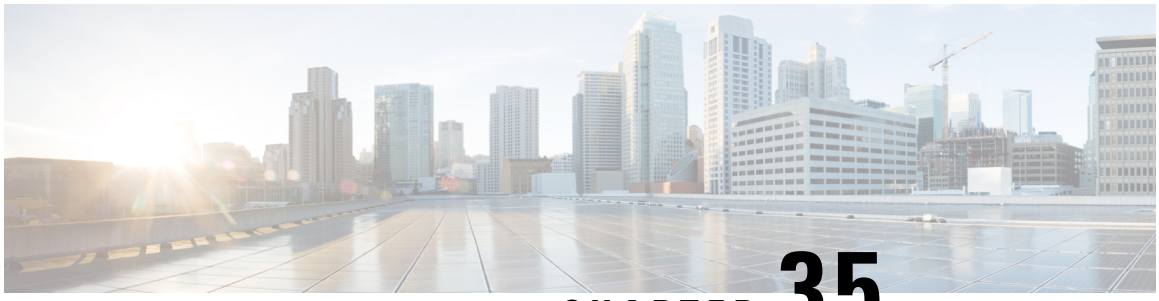
Feature Information for Standard IP Access List Logging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 56: Feature Information for Standard IP Access List Logging

Feature Name	Releases	Feature Information
Standard IP Access List Logging		The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list logs an information message about the packet at the device console.



CHAPTER 35

IP Access List Entry Sequence Numbering

The IP Access List Entry Sequence Numbering feature allows you to apply sequence numbers to **permit** or **deny** statements as well as reorder, add, or remove such statements from a named IP access list. The IP Access List Entry Sequence Numbering feature makes revising IP access lists much easier. Prior to this feature, you could add access list entries to the end of an access list only; therefore, needing to add statements anywhere except at the end of a named IP access list required reconfiguring the entire access list.

Restrictions for IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.
- This feature does not support old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list (NACL) configuration mode.

Information About IP Access List Entry Sequence Numbering

Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.
- Filter outgoing packets on an interface.
- Restrict the contents of routing updates.

- Limit debug output based on an address or protocol.
- Control virtual terminal line access.
- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.
- Trigger dial-on-demand routing (DDR) calls.

How an IP Access List Works

An access list is a sequential list consisting of a permit statement and a deny statement that apply to IP addresses and possibly upper-layer IP protocols. The access list has a name by which it is referenced. Many software commands accept an access list as part of their syntax.

An access list can be configured and named, but it is not in effect until the access list is referenced by a command that accepts an access list. Multiple commands can reference the same access list. An access list can control traffic arriving at the device or leaving the device, but not traffic originating at the device.

IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet and returns an Internet Control Message Protocol (ICMP) Host Unreachable message.
- If no conditions match, the packet is dropped. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.
- Only one access list per interface, per protocol, per direction is allowed.
- Inbound access lists process packets arriving at the device. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.

- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list**command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit**or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

- Before you add new ACL statements, provide time to the parser to clean up the deletion.

Source and Destination Addresses

Source and destination address fields in an IP packet are two typical fields on which to base an access list. Specify source addresses to control the packets being sent from certain networking devices or hosts. Specify destination addresses to control the packets being sent to certain networking devices or hosts.

Wildcard Mask and Implicit Wildcard Mask

When comparing the address bits in an access list entry to a packet being submitted to the access list, address filtering uses wildcard masking to determine whether to check or ignore the corresponding IP address bits. By carefully setting wildcard masks, an administrator can select one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value.
- A wildcard mask bit 1 means ignore that corresponding bit value.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes a default wildcard mask of 0.0.0.0.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

Transport Layer Information

You can filter packets based on transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP) or Internet Group Management Protocol (IGMP) packet.

Benefits IP Access List Entry Sequence Numbering

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry (statement) in the middle of an existing list, all of the entries *after* the desired position had to be removed. Then, once you added the new entry, you needed to reenter all of the entries you removed earlier. This method was cumbersome and error prone.

The IP Access List Entry Sequence Numbering feature allows you to add sequence numbers to access list entries and resequence them. When you add a new entry, you can choose the sequence number so that the entry is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced (reordered) to create room to insert the new entry.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If you enter an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If you enter an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If you enter a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Entries that contain a fully qualified 32-bit host address are hashed instead of linked. And entries that define a sub-net are maintained in a linked list that is sorted by the sequence number for speed of ACL classification. When a packet is matched against a standard ACL, the source address is hashed and matched against the hash table. If no match is found, it then searches the linked list for a possible match.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are always synchronized.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment from that number. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- The IP Access List Entry Sequence Numbering feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable.

How to Use Sequence Numbers in an IP Access List

Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. When completing this task, keep the following points in mind:

- Resequencing the access list entries is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates that functionality.
- In the following procedure, the **permit** command is shown in Step 5 and the **deny** command is shown in Step 6. However, that order can be reversed. Use the order that suits the need of your configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list** {**standard**| **extended**} *access-list-name*
5. Do one of the following:
   - *sequence-number* **permit** *source source-wildcard*
   - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Do one of the following:
   - *sequence-number* **deny** *source source-wildcard*
   - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Do one of the following:
   - *sequence-number* **permit** *source source-wildcard*
   - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
8. Do one of the following:
   - *sequence-number* **deny** *source source-wildcard*
   - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
9. Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.
10. **end**
11. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:**	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.

	Command or Action	Purpose
	`Device(config)# ip access-list resequence kmd1 100 15`	
Step 4	**ip access-list** {**standard**| **extended**} *access-list-name* **Example:** `Device(config)# ip access-list standard kmd1`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you subsequently specify **permit** and/or **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you subsequently specify **permit** and/or **deny** statements using the extended access list syntax.
Step 5	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended **permit** command syntax.
Step 6	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list uses a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended **deny** command syntax.
Step 7	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 permit tcp any any log`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the permit (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.

	Command or Action	Purpose
Step 8	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 deny tcp any any log`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the deny (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.
Step 9	Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.	Allows you to revise the access list.
Step 10	**end** **Example:** `Device(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 11	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list.

Examples

Review the output of the **show ip access-lists** command to see that the access list includes the new entries:

```
Device# show ip access-lists kmd1

Standard IP access list kmd1
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.0, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Configuration Examples for IP Access List Entry Sequence Numbering

Example: Resequencing Entries in an Access List

The following example shows access list resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values specified, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default the entry has a sequence number of 10 more than the last entry in the access list.

```
Device# show access-list 150

Extended IP access list 150
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any

Device(config)# ip access-list extended 150
Device(config)# ip access-list resequence 150 1 2
Device(config)# exit

Device# show access-list 150

Extended IP access list 150
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    10 permit tcp any any eq 22 log
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Example: Adding Entries with Sequence Numbers

In the following example, an new entry is added to a specified access list:

```
Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
```

```
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Device(config)# ip access-list standard tryon
Device(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Device(config-std-nacl)# exit
Device(config)# exit
Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Example: Entry Without Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Device(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Device(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255

Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Device(config-std-nacl)# end
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.0.0.0, wildcard bits 0.0.0.255
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Secure Shell	Configuring Secure Shell and Secure Shell Version 2 Support feature modules.
Configuring authentication and authorization	Configuring Authentication , Configuring Authorization , and Configuring Accounting feature modules.

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

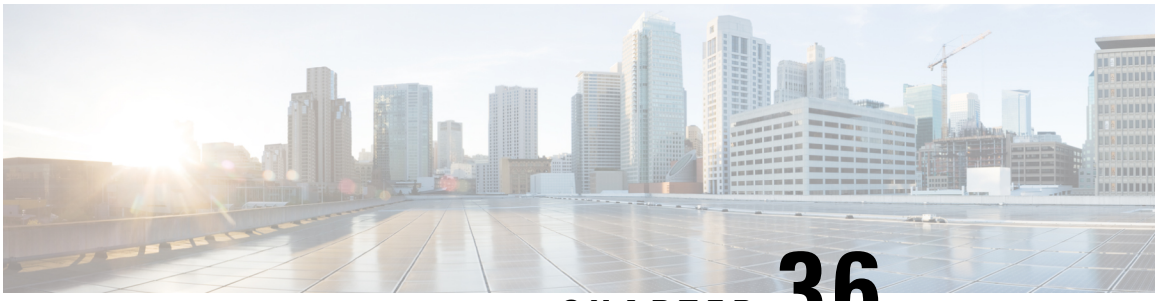
Feature Information for IP Access List Entry Sequence Numbering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 57: Feature Information for IP Access List Entry Sequence Numbering

Feature Name	Releases	Feature Information
IP Access List Entry Sequence Numbering		Users can apply sequence numbers to **permit** or **deny** statements and also reorder, add, or remove such statements from a named IP access list. This feature makes revising IP access lists much easier. Prior to this feature, users could add access list entries to the end of an access list only; therefore needing to add statements anywhere except the end required reconfiguring the access list entirely. In , , support was added for the Cisco Catalyst 3850 Series Switches. The following commands were introduced or modified: **deny (IP)**, **ip access-list resequence deny (IP), permit (IP)**.



CHAPTER 36

Configuring Lock-and-Key Security (Dynamic Access Lists)

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This chapter describes how to configure lock-and-key security at your router. Lock-and-key is a traffic filtering security feature available for the IP protocol.

For a complete description of lock-and-key commands, refer to the *Cisco IOS Security Command Reference* . To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Prerequisites for Configuring Lock-and-Key

Lock-and-key uses IP extended access lists. You must have a solid understanding of how access lists are used to filter traffic, before you attempt to configure lock-and-key. Access lists are described in the chapter "Access Control Lists: Overview and Guidelines."

Lock-and-key employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure lock-and-key. User authentication and authorization is explained in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Lock-and-key uses the **autocommand** command, which you should understand. This command is described in the *Cisco IOSTerminal Services Command Reference.*

Information About Configuring Lock-and-Key Security (Dynamic Access Lists)

About Lock-and-Key

Lock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists.

When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface's existing IP access list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first open a Telnet session to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and be able to reach their destination host.

Benefits of Lock-and-Key

Lock-and-key provides the same benefits as standard and static extended access lists (these benefits are discussed in the chapter "Access Control Lists: Overview and Guidelines"). However, lock-and-key also has the following security benefits over standard and static extended access lists:

- Lock-and-key uses a challenge mechanism to authenticate individual users.
- Lock-and-key provides simpler management in large internetworks.
- In many cases, lock-and-key reduces the amount of router processing required for access lists.
- Lock-and-key reduces the opportunity for network break-ins by network hackers.

With lock-and-key, you can specify which users are permitted access to which source and destination hosts. These users must pass a user authentication process before they are permitted access to their designated hosts. Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions.

When to Use Lock-and-Key

Two examples of when you might use lock-and-key follow:

- When you want a specific remote user (or group of remote users) to be able to access a host within your network, connecting from their remote hosts via the Internet. Lock-and-key authenticates the user, then permits limited access through your firewall router for the individual's host or subnet, for a finite period of time.
- When you want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local

user's hosts. Lock-and-key require the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts.

How Lock-and-Key Works

The following process describes the lock-and-key access operation:

1. A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router.
2. The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server.
3. When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.)
4. The user exchanges data through the firewall.
5. The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it. The configured timeout can either be an idle timeout or an absolute timeout.

Note The temporary access list entry is not automatically deleted when the user terminates a session. The temporary access list entry remains until a configured timeout is reached or until it is cleared by the system administrator.

Compatibility with Releases Before Cisco IOS Release 11.1

Enhancements to the **access-list** command are used for lock-and-key. These enhancements are backward compatible--if you migrate from a release before Cisco IOS Release 11.1 to a newer release, your access lists will be automatically converted to reflect the enhancements. However, if you try to use lock-and-key with a release before Cisco IOS Release 11.1, you might encounter problems as described in the following caution paragraph:

Caution Cisco IOS releases before Release 11.1 are not upwardly compatible with the lock-and-key access list enhancements. Therefore, if you save an access list with software older than Release 11.1, and then use this software, the resulting access list will not be interpreted correctly. This could cause you severe security problems. You must save your old configuration files with Cisco IOS Release 11.1 or later software before booting an image with these files.

Risk of Spoofing with Lock-and-Key

Caution Lock-and-key access allows an external event (a Telnet session) to place an opening in the firewall. While this opening exists, the router is susceptible to source address spoofing.

When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem.

To prevent spoofing, configure encryption so that traffic from the remote host is encrypted at a secured remote router, and decrypted locally at the router interface providing lock-and-key. You want to ensure that all traffic using lock-and-key will be encrypted when entering the router; this way no hackers can spoof the source address, because they will be unable to duplicate the encryption or to be authenticated as is a required part of the encryption setup process.

Router Performance Impacts with Lock-and-Key

When lock-and-key is configured, router performance can be affected in the following ways:

- When lock-and-key is triggered, the dynamic access list forces an access list rebuild on the silicon switching engine (SSE). This causes the SSE switching path to slow down momentarily.
- Dynamic access lists require the idle timeout facility (even if the timeout is left to default) and therefore cannot be SSE switched. These entries must be handled in the protocol fast-switching path.
- When remote users trigger lock-and-key at a border router, additional access list entries are created on the border router interface. The interface's access list will grow and shrink dynamically. Entries are dynamically removed from the list after either the idle-timeout or max-timeout period expires. Large access lists can degrade packet switching performance, so if you notice performance problems, you should look at the border router configuration to see if you should remove temporary access list entries generated by lock-and-key.

Maintaining Lock-and-Key

When lock-and-key is in use, dynamic access lists will dynamically grow and shrink as entries are added and deleted. You need to make sure that entries are being deleted in a timely way, because while entries exist, the risk of a spoofing attack is present. Also, the more entries there are, the bigger the router performance impact will be.

If you do not have an idle or absolute timeout configured, entries will remain in the dynamic access list until you manually remove them. If this is the case, make sure that you are extremely vigilant about removing entries.

Dynamic Access Lists

Use the following guidelines for configuring dynamic access lists:

- Do not create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do not assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol so that users must open a Telnet session into the router to be authenticated before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout--otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- If you realize that a job will run past the ACL's absolute timer, use the **access-list dynamic-extend** command to extend the absolute timer of the dynamic ACL by six minutes. This command allows you to open a new Telnet session into the router to re-authentication yourself using lock-and-key.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.
- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.
- To manually clear or to display dynamic access lists, refer to the section "Maintaining Lock-and-Key" later in this chapter.

Lock-and-Key Authentication

There are three possible methods to configure an authentication query process. These three methods are described in this section.

Note Cisco recommends that you use the TACACS+ server for your authentication query process. TACACS+ provides authentication, authorization, and accounting services. It also provides protocol support, protocol specification, and a centralized security database. Using a TACACS+ server is described in the next section, "Method 1--Configuring a Security Server."

Use a network access security server such as TACACS+ server. This method requires additional configuration steps on the TACACS+ server but allows for stricter authentication queries and more sophisticated tracking capabilities.

```
Router(config-line)# login tacacs
```

Use the **username** command. This method is more effective because authentication is determined on a user basis.

```
Router(config)# username

name
 {nopassword
 |
password
 {
mutual-password
 |
encryption-type

encryption-password
}}
```

Use the **password** and **login** commands. This method is less effective because the password is configured for the port, not for the user. Therefore, any user who knows the password can authenticate successfully.

```
R
outer(config-line)# password

password
Router(config-line)# login local
```

The autocommand Command

The **autocommand** command configures the system to automatically execute a specified privileged EXEC command when a user connects to a particular line. Use the following guidelines for configuring the **autocommand** command:

- If you use a TACACS+ server to authenticate the user, you should configure the **autocommand** command on the TACACS+ server as a per-user autocommand. If you use local authentication, use the **autocommand** command on the line.
- Configure all virtual terminal (VTY) ports with the same **autocommand** command. Omitting an **autocommand** command on a VTY port allows a random host to gain privileged EXEC mode access to the router and does not create a temporary access list entry in the dynamic access list.
- If you do not define an idle timeout with the **autocommand access-enable** command, you must define an absolute timeout with the **access-list** command. You must define either an idle timeout or an absolute timeout--otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated the session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure both idle and absolute timeouts, the absolute timeout value must be greater than the idle timeout value.

How to Configure Lock-and-Key Security (Dynamic Access Lists)

Configuring Lock-and-Key

To configure lock-and-key, use the following commands beginning in global configuration mode. While completing these steps, be sure to follow the guidelines listed in the "Lock-and-Key Configuration Guidelines" section of this chapter.

SUMMARY STEPS

1. Router(config)# **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **telnet** *source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
2. Router(config)# **access-list dynamic-extend**
3. Router(config)# **interface** *type number*
4. Router(config-if)# **ip access-group** *access-list-number*
5. Router(config-if)# **exit**
6. Router(config)# **line vty** *line-number* [*ending-line-number*]
7. Do one of the following:
   - Router(config-line)# **login tacacs**
   -
   - Router(config-line)# **password** *password*
8. Do one of the following:
   - Router(config-line)# **autocommand access-enable** [**host]** [**timeout** *minutes*]
   -
   - Router# **access-enable** [**host]** [**timeout** *minutes*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **telnet** *source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]	Configures a dynamic access list, which serves as a template and placeholder for temporary access list entries.
Step 2	Router(config)# **access-list dynamic-extend**	(Optional) Extends the absolute timer of the dynamic ACL by six minutes when you open another Telnet session into the router to re-authenticate yourself using lock-and-key. Use this command if your job will run past the ACL's absolute timer.

	Command or Action	Purpose
Step 3	Router(config)# **interface** *type number*	Configures an interface and enters interface configuration mode.
Step 4	Router(config-if)# **ip access-group** *access-list-number*	Applies the access list to the interface.
Step 5	Router(config-if)# **exit**	Exits interface configuration mode and enters global configuration mode.
Step 6	Router(config)# **line vty** *line-number* [*ending-line-number*]	Defines one or more virtual terminal (VTY) ports and enters line configuration mode. If you specify multiple VTY ports, they must all be configured identically because the software hunts for available VTY ports on a round-robin basis. If you do not want to configure all your VTY ports for lock-and-key access, you can specify a group of VTY ports for lock-and-key support only.

Step 7	Do one of the following:
	• Router(config-line)# **login tacacs**
	•
	• Router(config-line)# **password** *password*
	Example:
	`Router(config-line)# login local`
	Example:
	`Router(config-line)# exit`
	Example:
	`then`
	Example:
	`Router(config)# username name password secret`
		Configures user authentication in line or global configuration mode.

Step 8	Do one of the following: • Router(config-line)# **autocommand access-enable** [**host]** [**timeout** *minutes*] • • Router# **access-enable** [**host]** [**timeout** *minutes*]	Enables the creation of temporary access list entries in line configuration or privilege EXEC mode. Using the **autocommand** with the **access-enable** command in line configuration mode configures the system to automatically create a temporary access list entry in the dynamic access list when the host connects to the line (or lines). If the optional **host** keyword is not specified, all hosts on the entire network are allowed to set up a temporary access list entry. The dynamic access list contains the network mask to enable the new network connection. If the optional **timeout** keyword is specified, it defines the idle timeout for the temporary access list.

	Command or Action	Purpose
		Valid values, in minutes, range from 1 to 9999.

Verifying Lock-and-Key Configuration

You can verify that lock-and-key is successfully configured on the router by asking a user to test the connection. The user should be at a host that is permitted in the dynamic access list, and the user should have AAA authentication and authorization configured.

To test the connection, the user should Telnet to the router, allow the Telnet session to close, and then attempt to access a host on the other side of the router. This host must be one that is permitted by the dynamic access list. The user should access the host with an application that uses the IP protocol.

The following sample display illustrates what end-users might see if they are successfully authenticated. Notice that the Telnet connection is closed immediately after the password is entered and authenticated. The temporary access list entry is then created, and the host that initiated the Telnet session now has access inside the firewall.

```
Router% telnet corporate
Trying 172.21.52.1 ...
Connected to corporate.example.com.
Escape character is '^]'.
User Access Verification
Password:Connection closed by foreign host.
```

You can then use the **show access-lists** command at the router to view the dynamic access lists, which should include an additional entry permitting the user access through the router.s

Displaying Dynamic Access List Entries

You can display temporary access list entries when they are in use. After a temporary access list entry is cleared by you or by the absolute or idle timeout parameter, it can no longer be displayed. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show access-lists [access-list-number]`	Displays dynamic access lists and temporary access list entries.

Manually Deleting Dynamic Access List Entries

To manually delete a temporary access list entry, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear access-template [access-list-number | name] [dynamic-name] [source] [destination]`	Deletes a dynamic access list.

Configuration Examples for Lock-and-Key

Example Lock-and-Key with Local Authentication

This example shows how to configure lock-and-key access, with authentication occurring locally at the router. Lock-and-key is configured on the Ethernet 0 interface.

```
interface ethernet0
 ip address 172.18.23.9 255.255.255.0
 ip access-group 101 in
access-list 101 permit tcp any host 172.18.21.2 eq telnet
access-list 101 dynamic mytestlist timeout 120 permit ip any any
line vty 0
login local
autocommand access-enable timeout 5
```

The first access-list entry allows only Telnet into the router. The second access-list entry is always ignored until lock-and-key is triggered.

In the **access-list** command, the timeout is the absolute timeout. In this example, the lifetime of the mytestlist ACL is 120 minutes; that is, when a user logs in and enable the **access-enable** command, a dynamic ACL is created for 120 minutes (the maximum absolute time). The session is closed after 120 minutes, whether or not anyone is using it.

In the **access-enable**command, the timeout is the idle timeout. In this example, each time the user logs in or authenticates there is a 5-minute session. If there is no activity, the session closes in 5 minutes and the user has to reauthenticate. If the user uses the connection, the absolute time takes affect and the session closes in 120 minutes.

After a user opens a Telnet session into the router, the router will attempt to authenticate the user. If authentication is successful, the **autocommand** executes and the Telnet session terminates. The **autocommand** creates a temporary inbound access list entry at the Ethernet 0 interface, based on the second access-list entry (mytestlist). If there is no activity, this temporary entry will expire after 5 minutes, as specified by the timeout.

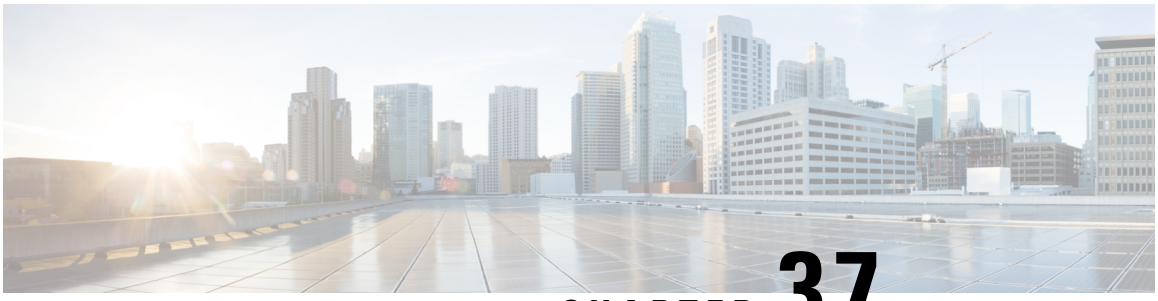
Example Lock-and-Key with TACACS+ Authentication

Cisco recommends that you use a TACACS+ server for authentication, as shown in the example.

The following example shows how to configure lock-and-key access, with authentication on a TACACS+ server. Lock-and-key access is configured on the BRI0 interface. Four VTY ports are defined with the password “password1”.

```
aaa authentication login default group tacacs+ enable
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable password ciscotac
!
isdn switch-type basic-dms100
!
interface ethernet0
ip address 172.18.23.9 255.255.255.0
!
interface BRI0
 ip address 172.18.21.1 255.255.255.0
```

```
 encapsulation ppp
 dialer idle-timeout 3600
 dialer wait-for-carrier-time 100
 dialer map ip 172.18.21.2 name dialermapname
 dialer-group 1
 isdn spid1 2036333715291
 isdn spid2 2036339371566
 ppp authentication chap
 ip access-group 102 in
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet
access-list 102 dynamic testlist timeout 5 permit ip any any
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2
priority-list 1 interface BRI0 high
tacacs-server host 172.18.23.21
tacacs-server host 172.18.23.14
tacacs-server key test1
tftp-server rom alias all
!
dialer-list 1 protocol ip permit
!
line con 0
 password password1
line aux 0
 line VTY 0 4
 autocommand access-enable timeout 5
 password password1
!
```



CHAPTER 37

ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options.

Restrictions for ACL IP Options Selective Drop

Resource Reservation Protocol (RSVP) (Multiprotocol Label Switching traffic engineering [MPLS TE]), Internet Group Management Protocol Version 2 (IGMPv2), and other protocols that use IP options packets may not function in drop or ignore modes.

Information About ACL IP Options Selective Drop

Using ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows a router to filter IP options packets, thereby mitigating the effects of these packets on a router and downstream routers, and perform the following actions:

- Drop all IP options packets that it receives and prevent options from going deeper into the network.
- Ignore IP options packets destined for the router and treat them as if they had no IP options.

For many users, dropping the packets is the best solution. However, in environments in which some IP options may be legitimate, reducing the load that the packets present on the routers is sufficient. Therefore, users may prefer to skip options processing on the router and forward the packet as though it were pure IP.

Benefits of Using ACL IP Options Selective Drop

- Drop mode filters packets from the network and relieves downstream routers and hosts of the load from options packets.
- Drop mode minimizes loads to the Route Processor (RP) for options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Now, the ignore and drop forms prevent the packets from impacting the RP performance.

How to Configure ACL IP Options Selective Drop

Configuring ACL IP Options Selective Drop

This section describes how to configure the ACL IP Options Selective Drop feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip options** {**drop** | **ignore**}
4. **exit**
5. **show ip traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip options** {**drop**	**ignore**} **Example:** `Router(config)# ip options drop`	Drops or ignores IP options packets that are sent to the router.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show ip traffic** **Example:** `Router# show ip traffic`	(Optional) Displays statistics about IP traffic.

Configuration Examples for ACL IP Options Selective Drop

Example Configuring ACL IP Options Selective Drop

The following example shows how to configure the router (and downstream routers) to drop all options packets that enter the network:

```
Router(config)# ip options drop
% Warning:RSVP and other protocols that use IP Options packets may not function in drop or
 ignore modes.
end
```

Example Verifying ACL IP Options Selective Drop

The following sample output is displayed after using the **ip options drop** command:

```
Router# show ip traffic
IP statistics:
  Rcvd:  428 total, 323 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other, 30 ignored
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragments, 0 couldn't fragment
  Bcast: 0 received, 0 sent
  Mcast: 323 received, 809 sent
  Sent:  809 generated, 591 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop, 0 unsupported-addr
         0 options denied, 0 source IP address zero
```

Additional References for IP Access List Entry Sequence Numbering

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"
IP access list commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

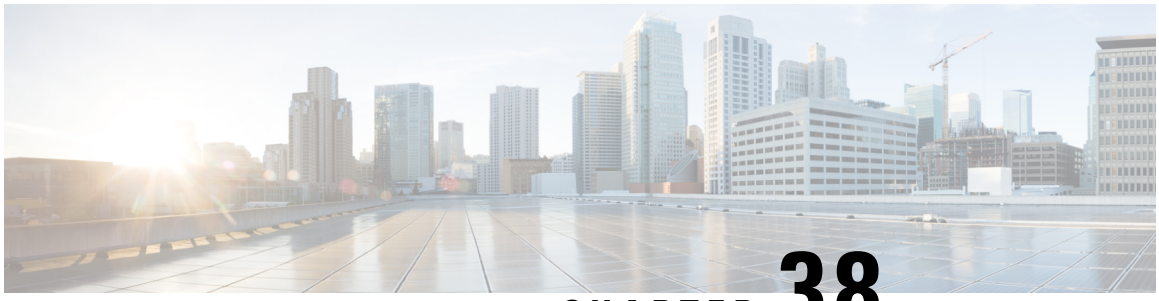
Feature Information for ACL IP Options Selective Drop

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 58: Feature Information for ACL IP Options Selective Drop

Feature Name	Releases	Feature Information
ACL IP Options Selective Drop	Cisco IOS XE Release 2.1	The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options. This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced: **ip options**.



CHAPTER 38

Displaying and Clearing IP Access List Data Using ACL Manageability

This module describes how to display the entries in an IP access list and the number of packets that have matched each entry. Users can get these statistics globally, or per interface and per incoming or outgoing traffic direction, by using the ACL Manageability feature. Viewing details of incoming and outgoing traffic patterns on various interfaces of a network device can help secure devices against attacks coming in on a particular interface. This module also describes how to clear counters so that the count of packets matching an access list entry will restart from zero.

Information About Displaying and Clearing IP Access List Data Using ACL Manageability

Benefits of ACL Manageability

Prior to Cisco IOS Release 12.4(6)T, the ACL infrastructure in Cisco IOS software maintained only global statistics for each ACE in an ACL. With this method, if an ACL is applied to multiple interfaces, the maintained ACE statistics are the sum of incoming and outgoing packet matches (hits) on all the interfaces on which that ACL is applied.

However, if ACE statistics are maintained per interface and per incoming or outgoing traffic direction, users can view specific details of incoming and outgoing traffic patterns and the effectiveness of ACEs on the various interfaces of a network device. This type of information is useful for securing devices against attacks coming in on a particular interface.

Support for Interface-Level ACL Statistics

With Cisco IOS Release 12.4(6)T, the ACL infrastructure in Cisco IOS software is now extended to support the maintenance, display, and clearing of ACE statistics per interface and per incoming or outgoing traffic direction for ACLs. This support is often referred to as "support for interface-level statistics."

Note If the same access-group ACL is also used by other features, the maintained interface statistics are not updated when a packet match is detected by the other features. In this case, the sum of all the interface level statistics that are maintained for an ACL may not add up to the global statistics for that ACL.

How to Display and Clear IP Access List Data

This section contains the following procedures for displaying IP access lists and the counts of packets that match (hit) each list, and for clearing IP access list counters.

Note Alternatively, if you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you. For more information, see the "IP Access List Logging" section of the "IP Access List Overview."

Displaying Global IP ACL Statistics

Perform this task to display all IP access lists on the router and counts of packets that have matched.

SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number* | *access-list-name*] **Example:** `Router# show ip access-list limited`	Displays IP access list information. • This example displays statistics for all interfaces that use the access list named "limited."

Displaying Interface-Level IP ACL Statistics

This section describes how to display IP ACE statistics per interface and per incoming or outgoing traffic direction for ACLs. This feature is known as ACL Manageability.

Note

- ACL Manageability supports:
  - Only nondistributed software switched platforms.
  - Standard and extended statically configured ACLs, and Threat Mitigation Service (TMS) dynamic ACEs.
- ACL Manageability does not support:
  - Reflexive and user-configured dynamic ACLs and dynamic ACE blocks, such as Firewall and Authentication Proxy.
  - Virtual-template and virtual-access interfaces.

>

SUMMARY STEPS

1. **enable**
2. **show ip access-list interface** *interface-name* [**in**| **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list interface** *interface-name* [**in**| **out**] **Example:** `Router# show ip access-list interface FastEthernet 0/0 in`	Displays IP access list information. • This example displays statistics about traffic coming into the FastEthernet interface. • To display debugging information about ACL interface-level statistics, use the **debug ip access-list intstats** command.

Clearing the Access List Counters

The system counts how many packets match (hit) each line of an access list; the counters are displayed by the **show access-lists** EXEC command. Perform this task to clear the counters of an access list. You might do this if you are trying to determine a more recent count of packets that match an access list, starting from zero.

SUMMARY STEPS

1. **enable**
2. **clear ip access-list counters** {*access-list-number* | *access-list-name}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip access-list counters** {*access-list-number* | *access-list-name}* **Example:** `Router# clear access-list counters corpmark`	Clears IP access list counters.

Configuration Examples for Displaying and Clearing IP Access List Data Using ACL Manageability

Example Displaying Global IP ACL Statistics

The following example displays global statistics for ACL 150:

```
Router# show ip access-list 150

Extended IP access list 150
   10 permit ip host 10.1.1.1 any (3 matches)
   30 permit ip host 10.2.2.2 any (27 matches)
```

Example Displaying Input Statistics

The following example displays statistics on incoming packets gathered from the FastEthernet interface 0/1, associated with access list 150 (ACL number):

```
Router#
 show ip access-list interface FastEthernet 0/1 in
Extended IP access list 150 in
   10 permit ip host 10.1.1.1 any (3 matches)
   30 permit ip host 10.2.2.2 any (12 matches)
```

Example Displaying Output Statistics

The following example displays statistics on outgoing packets gathered from the FastEthernet interface 0/0:

```
Router#
 show ip access-list interface FastEthernet 0/0 out
Extended IP access list myacl out
    5 deny ip any 10.1.0.0 0.0.255.255
    10 permit udp any any eq snmp (6 matches)
```

Example Displaying Input and Output Statistics

Note If no direction is specified, any input and output ACLs applied to that interface are displayed.

The following example displays input and output statistics gathered from the FastEthernet interface 0/0:

```
Router#
 show ip access-list interface FastEthernet 0/0
Extended IP access list 150 in
   10 permit ip host 10.1.1.1 any
   30 permit ip host 10.2.2.2 any (15 matches)
Extended IP access list myacl out
    5 deny ip any 10.1.0.0 0.0.255.255
    10 permit udp any any eq snmp (6 matches)
```

Example Clearing Global and Interface Statistics for an IP Access List

The following example clears global and interface statistics for IP ACL 150:

```
Router#
 clear ip access-list counters 150
```

Example Clearing Global and Interface Statistics for All IP Access Lists

The following example clears global and interface statistics for all IP ACLs:

```
Router#
 clear ip access-list counters
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

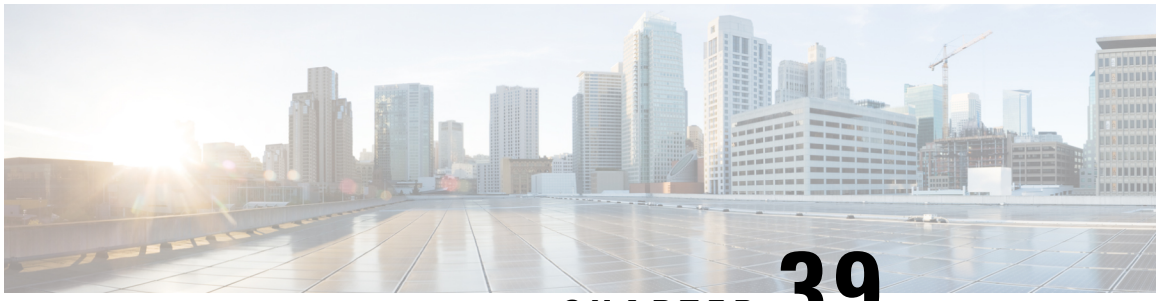
Feature Information for Displaying IP Access List Information and Clearing Counters

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 59: Feature Information for Displaying and Clearing IP Access List Data Using ACL Manageability

Feature Name	Releases	Feature Information
ACL Manageability	Cisco IOS XE Release 3.9S	The ACL Manageability feature enables users to display and clear Access Control Entry (ACE) statistics per interface and per incoming or outgoing traffic direction for access control lists (ACLs).



CHAPTER 39

ACL Syslog Correlation

The Access Control List (ACL) Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies the ACE , within the ACL, that generated the syslog entry.

Prerequisites for ACL Syslog Correlation

Before you configure the ACL Syslog Correlation feature, you must understand the concepts in the "IP Access List Overview" module.

The ACL Syslog Correlation feature appends a user-defined cookie or a device-generated hash value to ACE messages in the syslog. These values are only appended to ACE messages when the log option is enabled for the ACE.

Information About ACL Syslog Correlation

ACL Syslog Correlation Tags

The ACL Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies an ACE that generated the syslog entry.

Network management software can use the tag to identify which ACE generated a specific syslog event. For example, network administrators can select an ACE rule in the network management application and can then view the corresponding syslog events for that ACE rule.

To append a tag to the syslog message, the ACE that generates the syslog event must have the log option enabled. The system appends only one type of tag (either a user-defined cookie or a device-generated MD5 hash value) to each message.

To specify a user-defined cookie tag, the user must enter the cookie value when configuring the ACE log option. The cookie must be in alpha-numeric form, it cannot be greater than 64 characters, and it cannot start with hex-decimal notation (such as 0x).

To specify a device-generated MD5 hash value tag, the hash-generation mechanism must be enabled on the device and the user must not enter a cookie value while configuring the ACE log option.

ACE Syslog Messages

When a packet is matched against an access control entry (ACE) in an ACL, the system checks whether the log option is enabled for that event. If the log option is enabled and the ACL Syslog Correlation feature is configured on the device, the system attaches the tag to the syslog message. The tag is displayed at the end of the syslog message, in addition to the standard information.

The following is a sample syslog message showing a user-defined cookie tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) ->
192.168.16.2(23), 1 packet [User_permiited_ACE]
```

The following is a sample syslog message showing a hash value tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) ->
192.168.16.2(23), 1 packet [0x723E6E12]
```

How to Configure ACL Syslog Correlation

Enabling Hash Value Generation on a Device

Perform this task to configure the device to generate an MD5 hash value for each log-enabled access control entry (ACE) in the system that is not configured with a user-defined cookie.

When the hash value generation setting is enabled, the system checks all existing ACEs and generates a hash value for each ACE that requires one. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list logging hash-generation**
4. **end**
5. Do one of the following:
   - **show ip access-list** *access-list-number*
   - **show ip access-list** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list logging hash-generation** **Example:** `Device(config)# ip access-list logging hash-generation`	Enables hash value generation on the device. • If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Do one of the following:
	• **show ip access-list** *access-list-number*
	• **show ip access-list** *access-list-name*
	Example:
	`Device# show ip access-list 101`
	Example:
	`Device# show ip access-list acl`
		(Optional) Displays the contents of the numbered or named IP access list.
		• Review the output to confirm that the access list for a log-enabled ACE includes the generated hash value.

Disabling Hash Value Generation on a Device

Perform this task to disable hash value generation on the device. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip access-list logging hash-generation**
4. **end**
5. Do one of the following:
   - **show ip access-list** *access-list-number*

- **show ip access-list** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip access-list logging hash-generation** **Example:** `Device(config)# no ip access-list logging hash-generation`	Disables hash value generation on the device. • The system removes any previously created hash values from the system.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Do one of the following:
	• **show ip access-list** *access-list-number*
	• **show ip access-list** *access-list-name*
	Example:
	`Device# show ip access-list 101`
	Example:
	`Device# show ip access-list acl`
		(Optional) Displays the contents of the IP access list.
		• Review the output to confirm that the access list for a log-enabled ACE does not have a generated hash value.

Configuring ACL Syslog Correlation Using a User-Defined Cookie

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a user-defined cookie as the syslog message tag.

The example in this section shows how to configure the ACL Syslog Correlation feature using a user-defined cookie for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a user-defined cookie for both numbered and named access lists, and for both standard and extended access lists.

Note The following restrictions apply when choosing the user-defined cookie value:

- The maximum number of characters is 64.
- The cookie cannot start with hexadecimal notation (such as 0x).
- The cookie cannot be the same as, or a subset of, the following keywords: **reflect**, **fragment**, **time-range**. For example, reflect and ref are not valid values. However, the cookie can start with the keywords. For example, reflectedACE and fragment_33 are valid values
- The cookie must contains only alphanumeric characters.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **permit** *protocol source destination* **log** *word*
4. **end**
5. **show ip access-list** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **permit** *protocol source destination* **log** *word* **Example:** `Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log UserDefinedValue`	Defines an extended IP access list and a user-defined cookie value. • Enter the cookie value as the *word*argument.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip access-list** *access-list-number*	(Optional) Displays the contents of the IP access list.

	Command or Action	Purpose
	Example: `Device# show ip access-list 101`	• Review the output to confirm that the access list includes the user-defined cookie value.

Examples

The following is sample output from the **show ip access-list** command for an access list with a user-defined cookie value.

```
Device# show ip access-list
101
Extended IP access list 101
30 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = UserDefinedValue)
```

Configuring ACL Syslog Correlation Using a Hash Value

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a device-generated hash value as the syslog message tag.

The steps in this section shows how to configure the ACL Syslog Correlation feature using a device-generated hash value for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a device-generated hash value for both numbered and named access lists, and for both standard and extended access lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list logging hash-generation**
4. access-list *access-list-number* **permit** *protocol source destination* **log**
5. **end**
6. **show ip access-list** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list logging hash-generation**	Enables hash value generation on the device.

	Command or Action	Purpose
	Example: `Device(config)# ip access-list logging hash-generation`	• If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.
Step 4	access-list *access-list-number* **permit** *protocol source destination* **log** **Example:** `Device(config)# access-list 102 permit tcp host 10.1.1.1 host 10.1.1.2 log`	Defines an extended IP access list. • Enable the log option for the access list, but do not specify a cookie value. • The device automatically generates a hash value for the newly defined access list.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip access-list** *access-list-number* **Example:** `Device# show ip access-list 102`	(Optional) Displays the contents of the IP access list. • Review the output to confirm that the access list includes the router-generated hash value.

Examples

The following is sample output from the **show ip access-list** command for an access list with a device-generated hash value.

```
Device# show ip access-list
102
Extended IP access list 102
10 permit tcp host 10.1.1.1 host 10.1.1.2 log (hash = 0x7F9CF6B9)
```

Changing the ACL Syslog Correlation Tag Value

Perform this task to change the value of the user-defined cookie or replace a device-generated hash value with a user-defined cookie.

The steps in this section shows how to change the ACL Syslog Correlation tag value on a numbered access list. However, you can change the ACL Syslog Correlation tag value for both numbered and named access lists, and for both standard and extended access lists.

SUMMARY STEPS

1. **enable**
2. show access-list
3. **configure terminal**
4. access-list *access-list-number* **permit** *protocol source destination* **log** *word*

5. **end**
6. **show ip access-list** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show access-list **Example:** `Device(config)# show access-list`	(Optional) Displays the contents of the access list.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 4	access-list *access-list-number* **permit** *protocol source destination* **log** *word*
	Example:
	`Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV`
	Example:
	`OR`
	Example:
	Example:
	`Device(config)# access-list 101 permit tcp any any log replacehash`
		Modifies the cookie or changes the hash value to a cookie.
		• You must enter the entire access list configuration command, replacing the previous tag value with the new tag value.

Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip access-list** *access-list-number* **Example:** `Device# show ip access-list 101`	(Optional) Displays the contents of the IP access list. • Review the output to confirm the changes.

Troubleshooting Tips

Use the **debug ip access-list hash-generation** command to display access list debug information. The following is an example of the **debug** command output:

```
Device# debug ip access-list hash-generation
 Syslog hash code generation debugging is on
Device# show debug
IP ACL:
 Syslog hash code generation debugging is on
Device# no debug ip access-list hash-generation

 Syslog hash code generation debugging is off
Device# show debug
Device#
```

Configuration Examples for ACL Syslog Correlation

Example: Configuring ACL Syslog Correlation Using a User-Defined Cookie

The following example shows how to configure the ACL Syslog Correlation feature on a device using a user-defined cookie.

```
Device#
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.6 log cook_33_std
Device(config)# do show ip access 33
Standard IP access list 33
10 permit 10.10.10.6 log (tag = cook_33_std)
Device(config)# end
```

Example: Configuring ACL Syslog Correlation using a Hash Value

The following examples shows how to configure the ACL Syslog Correlation feature on a device using a device-generated hash value.

```
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.7 log
Device(config)#
*Nov 7 13:51:23.615: %IPACL-HASHGEN: Hash Input: 33 standard permit 10.10.10.7
Hash Output: 0xCE87F535
Device(config)#
do show ip access 33

Standard IP access list 33
    10 permit 10.10.10.6 log (tag = cook_33_std)
    20 permit 10.10.10.7 log (hash = 0xCE87F535)
```

Example: Changing the ACL Syslog Correlation Tag Value

The following example shows how to replace an existing access list user-defined cookie with a new cookie value, and how to replace a device-generated hash value with a user-defined cookie value.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# do show ip access-list 101
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = MyCookie)
    20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV
Device(config)# do show access-list
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
    20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp any any log replacehash
Device(config)# do show access-list
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
    20 permit tcp any any log (tag = replacehash)
```

Additional References for IPv6 IOS Firewall

Related Documents

Related Topic	Document Title
Security commands	• *Cisco IOS Security Command Reference: Commands A to C* • *Cisco IOS Security Command Reference: Commands D to L* • *Cisco IOS Security Command Reference: Commands M to R* • *Cisco IOS Security Command Reference: Commands S to Z*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

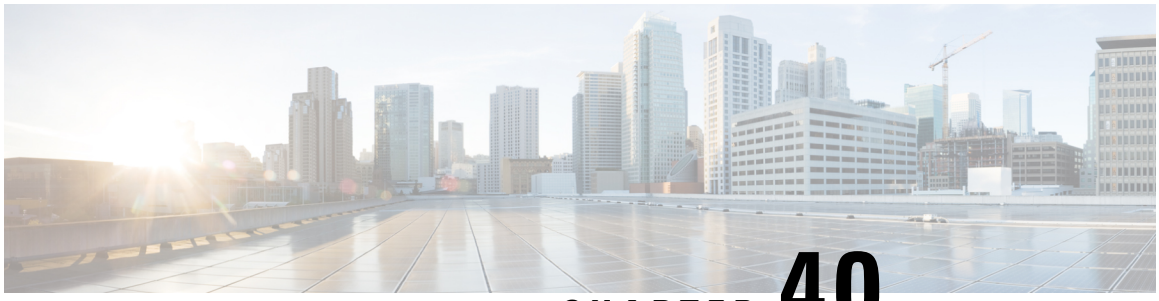
Feature Information for ACL Syslog Correlation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 60: Feature Information for ACL Syslog Correlation

Feature Name	Releases	Feature Information
ACL Syslog Correlation	Cisco IOS XE Release 3.6S	The ACL Syslog Correlation feature appends a tag (either a user-defined cookie or a router-generated MD5 hash value) to ACE syslog entries. This tag uniquely identifies the ACE , within the ACL, that generated the syslog entry.



CHAPTER 40

IPv6 Access Control Lists

Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering of traffic based on source and destination addresses, and inbound and outbound traffic to a specific interface. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.

This module describes how to configure IPv6 traffic filtering and to control access to virtual terminal lines.

RSP3 Porting Related Information

IPv6 ACL is not supported on RSP3

Information About IPv6 Access Control Lists

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list**command with the **deny** and **permit** keywords in global configuration mode.

IPv6 extended ACLs augments standard IPv6 ACL functionality to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

IPv6 Packet Inspection

The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local device address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local device address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

How to Configure IPv6 Access Control Lists

Configuring IPv6 Traffic Filtering

Creating and Configuring an IPv6 ACL for Traffic Filtering

Note IPv6 ACLs on the Cisco ASR 1000 platform do not contain implicit permit rules. The IPv6 neighbor discovery process uses the IPv6 network-layer service; therefore, to enable IPv6 neighbor discovery, you must add IPv6 ACLs to allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, uses a separate data-link-layer protocol; therefore, by default IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. Do one of the following:
   - **permit protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix / prefix-length* | **any** | **host** *destination-ipv6-address*} [**operator** [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
   - **deny** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*]

[**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list inbound`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode. • The *access-list name*argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
Step 4	Do one of the following: • **permit protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix / prefix-length* | **any** | **host** *destination-ipv6-address*} [**operator** [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] • **deny** *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport** **Example:** `Device(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any` **Example:**	Specifies permit or deny conditions for an IPv6 ACL.

	Command or Action	Purpose
	`Device(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input`	

Applying the IPv6 ACL to an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 traffic-filter** *access-list-name* {**in**| **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic-filter** *access-list-name* {**in**| **out**} **Example:** `Device(config-if)# ipv6 traffic-filter inbound in`	Applies the specified IPv6 access list to the interface specified in the previous step.

Controlling Access to a vty

Creating an IPv6 ACL to Provide Access Class Filtering

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*

4. Do one of the following:
   - **permit protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix / prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*
   - **deny** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list cisco`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.
Step 4	Do one of the following: • **permit protocol** {*source-ipv6-prefix/prefix-length*	**any**	**host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix / prefix-length*	**any**	**host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number*	*doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number*	*mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name* • **deny** *protocol* {*source-ipv6-prefix/prefix-length*	**any**	**host** *source-ipv6-address*} [*operator port-number*]] {*destination-ipv6-prefix/prefix-length*	**any**	**host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number*	*doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number*	*mh-type*]]	Specifies permit or deny conditions for an IPv6 ACL.

	Command or Action	Purpose

	[**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**
	Example:
	`Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:0:4::32 any`
	Example:
	`Device(config-ipv6-acl)# deny ipv6 host 2001:DB8:0:6::6 any`	

Applying an IPv6 ACL to the Virtual Terminal Line

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [**aux**| **console**| **tty**| **vty**] *line-number*[*ending-line-number*]
4. **ipv6 access-class** *ipv6-access-list-name* {**in**| **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line** [**aux**| **console**| **tty**| **vty**] *line-number*[*ending-line-number*] **Example:** `Device(config)# line vty 0 4`	Identifies a specific line for configuration and enters line configuration mode. • In this example, the **vty** keyword is used to specify the virtual terminal lines for remote console access.
Step 4	**ipv6 access-class** *ipv6-access-list-name* {**in**| **out**} **Example:** `Device(config-line)# ipv6 access-class cisco in`	Filters incoming and outgoing connections to and from the device based on an IPv6 ACL.

Configuration Examples for IPv6 Access Control Lists

Example: Verifying IPv6 ACL Configuration

In this example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Device> show ipv6 access-list

IPv6 access list inbound
    permit tcp any any eq bgp (8 matches) sequence 10
    permit tcp any any eq telnet  (15 matches) sequence 20
    permit udp any any  sequence 30

IPv6 access list Virtual-Access2.1#427819008151 (per-user)
    permit tcp host 2001:DB8:1::32 eq bgp host 2001:DB8:2::32 eq 11000 sequence 1
    permit tcp host 2001:DB8:1::32 eq telnet host 2001:DB8:2::32 eq 11001 sequence 2
```

Example: Creating and Applying an IPv6 ACL

The following example shows how to restrict HTTP access to certain hours during the day and log any activity outside of the permitted hours:

```
Device# configure terminal
Device(config)# time-range lunchtime
Device(config-time-range)# periodic weekdays 12:00 to 13:00
Device(config-time-range)# exit
Device(config)# ipv6 access-list INBOUND
Device(config-ipv6-acl)# permit tcp any any eq www time-range lunchtime
Device(config-ipv6-acl)# deny tcp any any eq www log-input
Device(config-ipv6-acl)# permit tcp 2001:DB8::/32 any
Device(config-ipv6-acl)# permit udp 2001:DB8::/32 any
Device(config-ipv6-acl)# end
```

Example: Controlling Access to a vty

In the following example, incoming connections to the virtual terminal lines 0 to 4 are filtered based on the IPv6 access list named acl1:

```
ipv6 access-list acl1
 permit ipv6 host 2001:DB8:0:4::2/32 any
!
line vty 0 4
 ipv6 access-class acl1 in
```

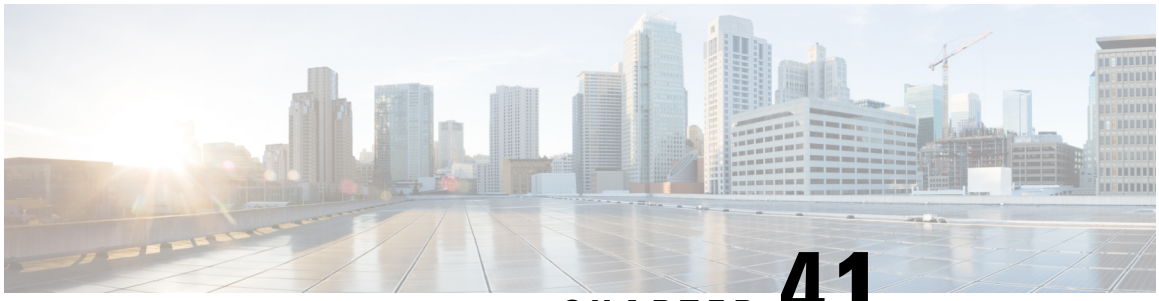
Feature Information for IPv6 Access Control Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 61: Feature Information for IPv6 Access Control Lists

Feature Name	Releases	Feature Information
IPv6 Services: Extended Access Control Lists	Cisco IOS XE Release 2.1	Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.



CHAPTER 41

IPv6 ACL Undetermined-Transport Support

The IPv6 ACL Undetermined-Transport Support feature helps in dropping misconfigured packets where the complete upper layer header is not present.

Restrictions for IPv6 ACL Undetermined-Transport Support

- The undetermined-transport option is supported only for Cisco Application Control Engines (ACE) with deny action and IPv6 protocol.
- Undetermined transport is not applied on nonfirst fragment packets.

Information about IPv6 ACL Undetermined-Transport Support

IPv6 ACL Undetermined-Transport

Unintended misconfigurations by users or malicious attacks on the network may cause operational problems for hosts on the network.

Upper layer header is placed at the end of Extended Header (EH) chain in IPv6 packet, as it described in RFC 2460. If the complete upper layer header is not present in the IPv6 packet, then the router cannot process the packet. These packets may be misconfigured, corrupted, or malicious packets.

You may choose to drop these packets using IPv6 ACL with undetermined-transport option.

How to Configure IPv6 ACL Undetermined-Transport Support

Configuring IPv6 ACL Undetermined-Transport Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *acl-name*
4. **deny ipv6** {*src-addr* | **any**} {*dest-addr* | **any**} [**undetermined-transport**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *acl-name* **Example:** `Device(config)# ipv6 access-list acl1`	Configures IPv6 access list.
Step 4	**deny ipv6** {*src-addr* | **any**} {*dest-addr* | **any**} [**undetermined-transport**] **Example:** `Device(config-ipv6-acl)# deny ipv6 2001:DB8:0300:0201::/32 2001:DB8:1:1::/64 undetermined-transport`	Sets deny condition for an IPv6 access list as undermined transport.
Step 5	**end** **Example:** `Device(config-ipv6-acl)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 ACL Undetermined-Transport Support

Example: Example for IPv6 ACL Undetermined-Transport Support

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list acl1
Device(config-ipv6-acl)# deny ipv6 2001:DB8:0300:0201::/32 2001:DB8:1:1::/64
undetermined-transport
Device(config-ipv6-acl)# end
```

Additional References for IPv6 ACL Undetermined-Transport Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP access list commands	*Cisco IOS Security Command Reference*
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

Standards and RFCs

Standards/RFCs	Title
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

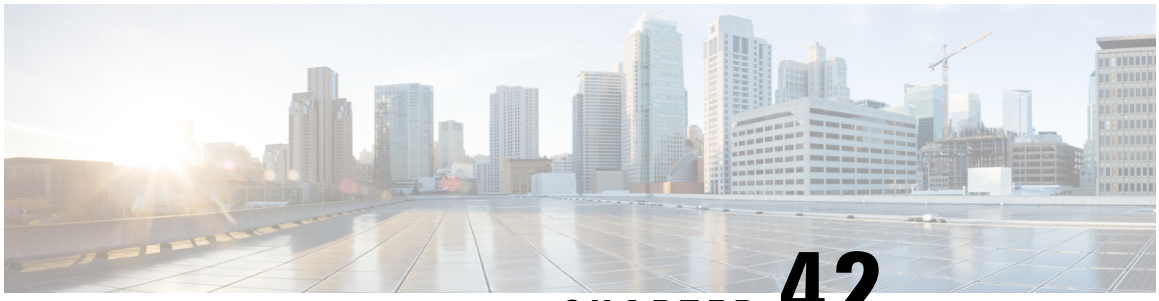
Feature Information for ACL Template

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 62: Feature Information for ACL Template

Feature Name	Releases	Feature Information
IPv6 ACL Undetermined-Transport Support	Cisco IOS XE Release 3.15	The IPv6 ACL Undetermined-Transport Support feature helps in dropping misconfigured packets, where the complete upper layer header is not present. No commands were introduced or modified.



CHAPTER 42

Configuring Template ACLs

When user profiles are configured using RADIUS Attribute 242 or vendor-specific attribute (VSA) Cisco-AVPairs, similar per-user access control lists (ACLs) may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

In networks where each subscriber has its own ACL, it is common for the ACL to be the same for each user except for the user's IP address. The Template ACLs feature groups ACLs with many common access control elements (ACEs) into a single ACL that saves system resources.

Prerequisites for Template ACLs

- Cisco ASR 1000 series routers
- Cisco IOS XE Release 2.4 or a later release

Restrictions for Template ACLs

Template ACLs are activated only for per-user ACLs configured through RADIUS Attribute 242 or VSA Cisco-AVPairs (ip:inacl/outacl). No other ACL types are processed by the Template ACL feature.

Template ACL functionality is available only for IPv4 ACLs.

Template ACL functionality is not available for the following types of per-user ACLs:

- Time-based ACLs
- Dynamic ACLs

- Evaluate ACLs
- Reflexive ACLs
- ACLs configured on ISG IP sessions
- IPv6 ACLs

Disabling the Template ACL Feature

When the Template ACL feature is disabled, the system replaces all existing template ACL instances with ACLs. If the system does not have enough resources (in particular TCAM resources) to setup the required number of ACLs, the system generates an error message, and the request to disable the Template ACLs feature fails.

Information About Configuring Template ACLs

Template ACL Feature Design

When the service provider uses AAA servers to configure individual ACLs for each authorized session using with RADIUS attribute 242 or VSA Cisco-AVPairs, the number of sessions can easily exceed the maximum ACL number allowed by the system.

In networks where each subscriber has an ACL, it is common for the ACL to be the same for each user except for the user's IP address. Template ACLs alleviate this problem by grouping ACLs with many common ACEs into a single ACL that compiles faster and saves system resources.

The Template ACL feature is enabled by default, and ACLs set up using the RADIUS attribute 242 or VSA Cisco-AVPairs are considered for template status.

When the Template ACL feature is enabled, the system scans and evaluates all configured per-session ACLs and then creates all required template ACLs.

Disabling Template ACLs

When the Template ACL feature is disabled, the system replaces all existing template ACL instances with ACLs. If the system does not have enough resources (in particular TCAM resources) to setup the required number of ACLs, the system generates an error message, and the request to disable the Template ACL feature fails.

Therefore, before you disable the Template ACL feature, use the **show access-list template summary** command to view the number of template ACLs in the system and ascertain if this number exceeds the system limitations.

When the template ACL feature is disabled, no new ACLS are considered for templating.

Multiple ACLs

When the Template ACL feature is enabled, the system can identify when two per-user ACLS are similar, and the system consolidates the two per-user ACLs into one template ACL.

For example, the following example shows two ACLs for two separate users:

```
ip access-list extended Virtual-Access1.1#1 (PeerIP: 10.1.1.1)
permit igmp any host 10.1.1.1
permit icmp host 10.1.1.1 any
deny ip host 10.31.66.36 host 10.1.1.1
deny tcp host 10.1.1.1 host 10.31.66.36
permit udp any host 10.1.1.1
permit udp host 10.1.1.1 any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
ip access-list extended Virtual-Access1.1#2 (PeerIP: 10.13.11.2)
permit igmp any host 10.13.11.2
permit icmp host 10.13.11.2 any
deny ip host 10.31.66.36 host 10.13.11.2
deny tcp host 10.13.11.2 host 10.31.66.36
permit udp any host 10.13.11.2
permit udp host 10.13.11.2 any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
```

With the Template ACL feature is enabled, the system recognizes that these two ACLs are similar, and creates a template ACL as follows:

```
ip access-list extended Template_1
permit igmp any host <PeerIP>
permit icmp host <PeerIP> any
deny ip host 10.31.66.36 host <PeerIP>
deny tcp host <PeerIP> 10.31.66.36
permit udp any host <PeerIP>
permit udp host <PeerIP> any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
```

In this example, the peer IP address is associated as follows:

- Virtual-Access1.1#1 10.1.1.1
- Virtual-Access1.1#2 10.13.11.2

The two ACLs are consolidated into one template ACL and are referenced as follows:

Virtual-Access1.1#1 maps to Template_1(10.1.1.1)

Virtual-Access1.1#2 maps to Template_1(10.13.11.2)

VSA Cisco-AVPairs

Template ACL processing occurs for ACLs that are configured using Cisco-AVPairs. Only AVPairs that are defined using the ACL number are considered for the templating process.

To be considered for templating, AVPairs for incoming ACLs must conform to the following format:

ip:inacl#number={standard-access-control-list | extended-access-control-list}

For example: ip:inacl#10=deny ip any 10.13.16.0 0.0.0.255

To be considered for templating, AVPairs for outgoing ACLs must conform to the following format:

ip:outacl#number={standard-access-control-list | extended-access-control-list}

For example: ip:outacl#200=permit ip any any

For more information on Cisco-AVPairs, see the Cisco Vendor-Specific AVPair Attributes section of the *Cisco IOS ISG RADIUS CoA Interface Guide*.

RADIUS Attribute 242

Template ACL processing occurs for ACLs that are configured using RADIUS attribute 242. Attribute 242 has the following format for an IP data filter:

Ascend-Data-Filter = "ip <dir> <action> [dstip <dest_ipaddr\subnet_mask>] [srcp <src_ipaddr\subnet_mask>] [<proto> [dstport <cmp> <value>] [srcport <cmp> <value>] [<est>]]"

The table below describes the elements in an attribute 242 entry for an IP data filter.

Table 63: IP Data Filter Syntax Elements

Element	Description
ip	Specifies an IP filter.
<dir>	Specifies the filter direction. Possible values are **in** (filtering packets coming into the router) or **out** (filtering packets going out of the router).
<action>	Specifies the action the router should take with a packet that matches the filter. Possible values are **forward** or **drop**.
dstip <dest_ipaddr\subnet_mask>	Enables destination-IP-address filtering. Applies to packets whose destination address matches the value of **<dest_ipaddr>**. If a subnet mask portion of the address is present, the router compares only the masked bits. If you set **<dest_ipaddr>** to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
srcp<src_ipaddr\subnet_mask>	Enables source-IP-address filtering. Applies to packets whose source address matches the value of **<src_ipaddr>**. If a subnet mask portion of the address is present, the router compares only the masked bits. If you set **<src_ipaddr>** to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
<proto>	Specifies a protocol specified as a name or a number. Applies to packets whose protocol field matches this value. Possible names and numbers are **icmp** (**1**), **tcp** (**6**), **udp** (**17**), and **ospf** (**89**). If you set this value to zero (0), the filter matches any protocol.

Element	Description
dstport <cmp> <value>	Enables destination-port filtering. This keyword is valid only when **<proto>** is set to **tcp** (**6**) or **udp** (**17**). If you do not specify a destination port, the filter matches any port. **<cmp>** defines how to compare the specified **<value>** to the actual destination port. This value can be <, =, >, or **!**. **<value>** can be a name or a number. Possible names and numbers are **ftp-data (20)**, **ftp (21)**, **telnet (23)**, **nameserver (42)**, **domain (53)**, **tftp (69)**, **gopher (70)**, **finger (79)**, **www (80)**, **kerberos (88)**, **hostname (101)**, **nntp (119)**, **ntp (123)**, **exec (512)**, **login (513)**, **cmd (514)**, and **talk (517)**.
srcport <cmp> <value>	Enables source-port filtering. This keyword is valid only when **<proto>** is set to **tcp**(**6**)or **udp** (**17**). If you do not specify a source port, the filter matches any port. **<cmp>** defines how to compare the specified **<value>** to the actual destination port. This value can be <, =, >, or **!**. **<value>** can be a name or a number. Possible names and numbers are **ftp-data** (**20**), **ftp** (**21**), **telnet**(**23**), **nameserver**(**42**), **domain**(**53**), **tftp**(**69**), **gopher**(**70**), **finger**(**79**), **www**(**80**), **kerberos** (**88**), **hostname** (**101**), **nntp** (**119**), **ntp**(**123**), **exec** (**512**), **login** (**513**), **cmd** (**514**), and **talk** (**517**).
<est>	When set to 1, specifies that the filter matches a packet only if a TCP session is already established. This argument is valid only when **<proto>** is set to **tcp (6)**.

"RADIUS Attribute 242 IP Data Filter Entries" shows four attribute 242 IP data filter entries.

RADIUS Attribute 242 IP Data Filter Entries

```
Ascend-Data-Filter="ip in drop"
Ascend-Data-Filter="ip out forward tcp"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16
dstport!=telnet"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16 icmp"
```

How to Configure Template ACLs

If ACLs are configured using RADIUS Attribute 242 or VSA Cisco-AVPairs, template ACLs are enabled by default.

Configuring the Maximum Size of Template ACLs

By default, template ACL status is limited to ACLs with 100 or fewer rules. However, you can set this limit to a lower number. To set the maximum number of rules that an ACL may have in order to be considered as a template ACL, perform the steps in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list template** *number*

4. **exit**
5. **show access-list template summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list template** *number* **Example:** `Router(config)# access-list template 50`	Enables template ACL processing. Only ACLs with the specified number of rules (or fewer rules) will be considered for template status.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show access-list template summary** **Example:** `Router# show access-list template summary`	(Optional) Displays summary information about template ACLs.

Troubleshooting Tips

The following commands can be used to troubleshoot the Template ACL feature:

- **show access-list template**
- show platform hardware qfp active classification class-group-manager class-group client acl all
- **show platform hardware qfp active feature acl** {**control** | **node** *acl-node-id*}
- show platform software access-list

Configuration Examples for Template ACLs

Example Maximum Size of Template ACLs

The following example shows how to set the maximum number of rules that an ACL may have in order to be considered for template status to 50. Only ACLs whose number of rules is the same as or smaller than 50 are considered for template status.

```
Router> enable

Router# configure terminal

Router(config)# access-list template 50
Router(config)# exit
```

Example Showing ACL Template Summary Information

The following example shows how to view summary information for all ACLs in the system. The output from the command includes the following information:

- Maximum number of rules per template ACL
- Number of discovered active templates
- Number of ACLs replaced by those templates
- Number of elements in the Red-Black tree

```
Router# show access-list template summary
Maximum rules per template ACL = 100
Templates active = 9
Number of ACLs those templates represent = 14769
Number of tree elements = 13
```

Red-Black Tree Elements

The number of tree elements is the number of elements in the Red-Black tree. Each template has 1 unique entry in the Red-Black tree. The system calculates a cyclic redundancy check (CRC) over each ACL masking out the peer IP address and puts the CRC into the Red-Black tree. For example:

Your system has 9 templates (representing 14769 ACLs), and 13 tree elements. If each template has only 1 unique entry in the Red-Black tree, then the additional 4 tree elements means that your system contains 4 per-user ACLs that are not templated.

Example Showing ACL Template Tree Information

The following example shows how to view Red-Black tree information for all ACLs in the system.

The output from the command includes the following information:

- Name of the ACL on the Red-Black tree

- The original CRC32 value
- Number of users of the ACL
- Calculated CRC32 value

```
Router# show access-list template tree
ACL name      OrigCRC   Count  CalcCRC
4Temp_1073741891108      59DAB725   98  59DAB725
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Secure Shell	Configuring Secure Shell and Secure Shell Version 2 Support feature modules.
Configuring authentication and authorization	Configuring Authentication , Configuring Authorization , and Configuring Accounting feature modules.

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

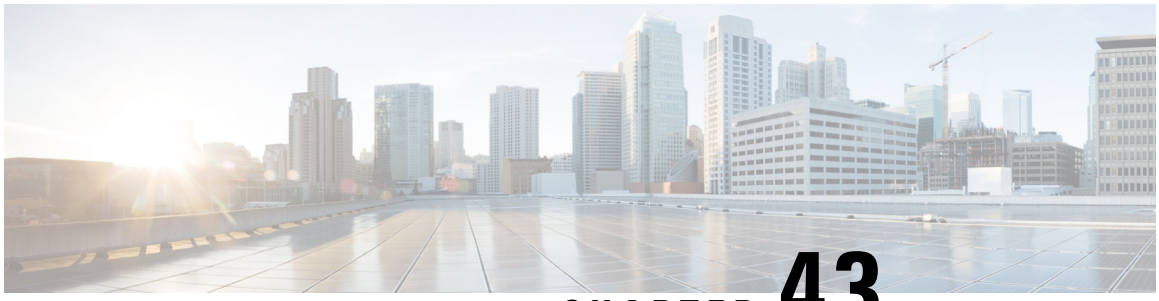
Feature Information for ACL Template

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 64: Feature Information for ACL Template

Feature Name	Releases	Feature Information
Template ACLs	12.2(28)SB 12.2(31)SB2 Cisco IOS XE Release 2.4	In 12.2(28)SB, this feature was introduced on the Cisco 10000 series router. In 12.2(31)SB2, support was added for the PRE3. In Cisco IOS XE Release 2.4, this feature was implemented on the Cisco ASR 1000 series routers. The following commands were introduced or modified:**access-list template, show access-list template**



CHAPTER 43

IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.

Information About IPv6 ACL—Template ACL

IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.

How to Enable IPv6 ACL—Template ACL

Enabling IPv6 Template Processing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list template** [*number-of-rules*]
4. **exit**
5. **show access-list template** {**summary** | *aclname* | **exceed** *number* | **tree**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list template** [*number-of-rules*] **Example:** `Router(config)# access-list template 50`	Enables template ACL processing. • The example in this task specifies that ACLs with 50 or fewer rules will be considered for template ACL status. • The *number-of-rules* argument default is 100.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and places the router in privileged EXEC mode.
Step 5	**show access-list template** {**summary** | *aclname* | **exceed** *number* | **tree**} **Example:** `Router# show access-list template summary`	Displays information about ACL templates.

Configuration Examples for IPv6 ACL—Template ACL

Example: IPv6 Template ACL Processing

In this example, the contents of ACL1 and ACL2 are the same, but the names are different:

```
ipv6 access-list extended ACL1 (PeerIP: 2001:1::1/64)
permit igmp any                 2003:1::1/64
permit icmp 2002:5::B/64        any
permit udp  any                 host 2004:1::5
permit udp  any                 host 2002:2BC::a
permit icmp host 2001:BC::7     host 2003:3::7
ipv6 access-list extended ACL2 (PeerIP: 2007:2::7/64)
permit igmp any                 2003:1::1/64
permit icmp 2002:5::B/64        any
permit udp  any                 host 2004:1::5
permit udp  any                 host 2002:2BC::a
permit icmp host 2001:BC::7     host 2003:3::7
```

The template for these ACLs is as follows:

```
ipv6 access-list extended Template_1
permit igmp any                 2003:1::1/64
permit icmp 2002:5::B/64        any
permit udp  any                 host 2004:1::5
permit udp  any                 host 2002:2BC::a
permit icmp host 2001:BC::7     host 2003:3::7
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*

Related Topic	Document Title
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

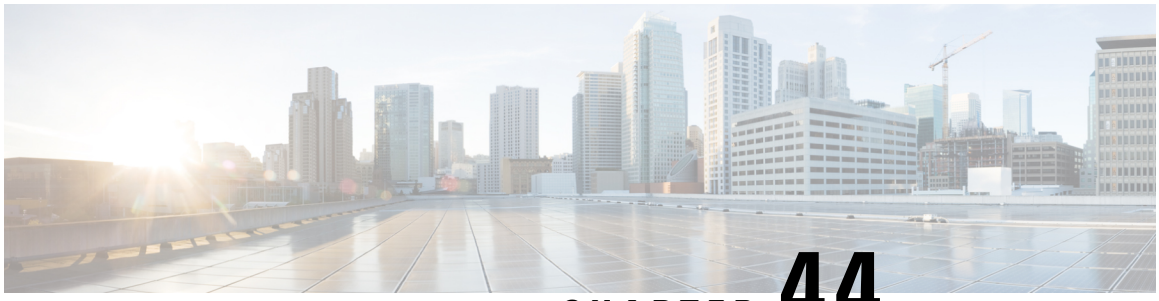
Feature Information for IPv6 ACL—Template ACL

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 65: Feature Information for IPv6 ACL—Template ACL

Feature Name	Releases	Feature Information
IPv6 ACL—Template ACL	Cisco IOS XE Release 3.2S	This feature allows similar per-user IPv6 ACLs to be replaced by a single template ACL. The following commands were introduced or modified: **access-list template**, **show access-list template**.



CHAPTER 44

IPv4 ACL Chaining Support

ACL Chaining, also known as Multi-Access Control List, allows you to split access control lists (ACLs). This module describes how with the IPv4 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

Restrictions for IPv4 ACL Chaining Support

- A single access control List (ACL) cannot be used for both common and regular ACLs for the same target in the same direction.
- ACL chaining applies to only security ACLs. It is not supported for feature policies, such as Quality of Service (QoS), Firewall Services Module (FW) and Policy Based Routing (PBR).
- Per-target statistics are not supported for common ACLs.

Information About IPv4 ACL Chaining Support

ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs

- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

IPv4 ACL Chaining Support

IPv4 ACL Chaining Support allows you to split the Access control list (ACL) into common and customer-specific ACLs and attach both ACLs to a common session. In this way, only one copy of the common ACL is attached to Ternary Content Addressable Memory (TCAM) and shared by all users, thereby making it easier to maintain the common ACEs.

The IPv4 ACL Chaining feature allows two IPV4 ACLs to be active on an interface per direction:

- Common
- Regular
- Common and Regular

Note If you configure both common and regular ACLs on an interface, the common ACL is considered over a regular ACL.

How to Configure IPv4 ACL Chaining Support

ACL chaining is supported by extending the **ip traffic filter** command.

The **ip traffic filter** command is not additive. When you use this command, it replaces earlier instances of the command.

For more information, refer to the *IPv6 ACL Chaining with a Common ACL* section in the Security Configuration Guide: Access Control Lists Configuration Guide.

Configuring an Interface to Accept Common ACL

Perform this task to configure the interface to accept a common Access control list (ACL) along with an interface-specific ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*}
4. **ip access-group** {**common** {*common-access-list-name* {*regular-access-list* | **acl**}}{**in** | **out**}}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number*} **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface (in this case a gigabitethernet interface) and enters the interface configuration mode.
Step 4	**ip access-group** {**common** {*common-access-list-name* {*regular-access-list* | **acl**}}{**in** | **out**}} **Example:** `Device(config)# ipv4 access-group common acl-p acl1 in`	Configures the interface to accept a common ACL along with the interface-specific ACL.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv4 ACL Chaining Support

This section provides configuration examples of Common Access Control List (ACL).

Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an Access Control List (ACL) configured on the interface without explicitly deleting the ACL:

```
interface gigabitethernet 0/0/0
ipv4 access-group common  C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface gigabitethernet 0/0/0
ipv4 access-group common  C_acl ACL2 in
end
```

This example shows how common ACL cannot be replaced on interfaces without deleting it explicitly from the interface:

```
interface gigabitethernet 0/0/0
ipv4 access-group common  C_acl1 ACL1 in
end
change the common acl to C_acl2
interface gigabitethernet 0/0/0
no ipv4 access-group common  C_acl1  ACL1 in
end
interface gigabitethernet 0/0/0
ipv4 access-group common  C_acl2  ACL1 in
end
```

Note When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.

Note If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how the interface ACL is removed:

```
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl1 ACL1 in
end
```

Additional References for IPv4 ACL Chaining Support

Related Documents

Related Topic	Document Title
IPv6 ACL Chaining Support	
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

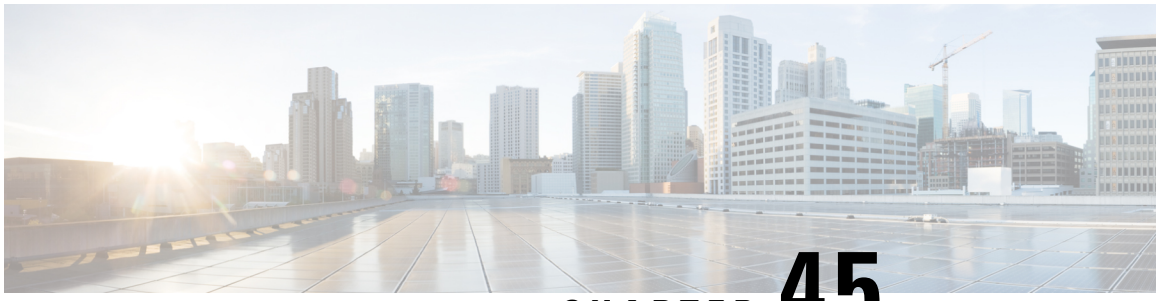
Feature Information for IPv4 ACL Chaining Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 66: Feature Information for IPv4 ACL Chaining Support

Feature Name	Releases	Feature Information
IPv4 ACL Chaining Support	Cisco IOS XE Release 3.11S Cisco IOS XE Release 3.6E	The IPv4 ACL Chaining Support feature describes how you can explicitly split Access control lists (ACLs) into common and user-specific ACLs and bind both ACLs to a session for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage. The following commands were introduced or modified: **ip access-group command**.



CHAPTER 45

IPv6 ACL Chaining with a Common ACL

ACL Chaining, also known as Multi-Access Control List (ACL), allows you to split ACLs. This document describes how with the IPv6 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

Information About IPv6 ACL Chaining with a Common ACL

ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs
- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

IPv6 ACL Chaining with a Common ACL

With IPv6 ACL Chaining, you can configure a traffic filter with the following:

- Common ACL
- Specific ACL
- Common and Specific ACL

Each Access control list (ACL) is matched in a sequence. For example, if you have specified both the ACLs - a common and a specific ACL, the packet is first matched against the common ACL; if a match is not found, it is then matched against the specific ACL.

Note Any IPv6 ACL may be configured on a traffic filter as a common or specific ACL. However, the same ACL cannot be specified on the same traffic filter as both common and specific.

How to Configure IPv6 ACL Chaining with a Common ACL

Before you begin

IPv6 ACL chaining is configured on an interface using an extension of the existing IPv6 traffic-filter command: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] [**in** | **out**]

Note You may choose to configure either of the following:

- Only a common ACL. For example: **ipv6 traffic-filter common** *common-acl*
- Only a specific ACL. For example: **ipv6 traffic-filter** *common-acl*
- Both ACLs. For example: **ipv6 traffic-filter common** *common-acl specific-acl*

The ipv6 traffic-filter command is not additive. When you use the command, it replaces earlier instances of the command. For example, the command sequence: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] **in ipv6 traffic-filter** [*specific-acl*] **in** binds a common ACL to the traffic filter, removes the common ACL and then binds a specific ACL.

Configuring the IPv6 ACL to an Interface

Perform this task to configure the interface to accept a common access control list (ACL) along with an interface-specific ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*}
4. **ipv6 traffic filter** {*common-access-list-name* {**in** | **out**}}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number*} **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic filter** {*common-access-list-name* {**in** | **out**}} **Example:** `Device(config)# ipv6 traffic-filter outbound out`	Applies the specified IPv6 access list to the interface specified in the previous step.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 ACL Chaining with a Common ACL

You may configure the following combinations in no particular order:

- A common ACL, for example: **ipv6 traffic-filter common** *common-acl* **in**
- A specific ACL, for example: **ipv6 traffic-filter** *specific-acl* **in**
- Both ACLs, for example: **ipv6 traffic-filter common** *common-acl* *specific-acl* **in**

Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an access control list (ACL) configured on the interface without explicitly deleting the ACL:

```
interface gigabitethernet 0/0/0
ipv6 access-group common  C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface gigabitethernet 0/0/0
ipv6 access-group common  C_acl ACL2 in
```

```
end
```

This example shows how to delete a common ACL from an interface. A common ACL cannot be replaced on interfaces without deleting it explicitly from the interface.

```
interface gigabitethernet 0/0/0
ipv6 access-group common  C_acl1 ACL1 in
end
change the common acl to C_acl2
interface gigabitethernet 0/0/0
no ipv6 access-group common  C_acl1  ACL1 in
end
interface gigabitethernet 0/0/0
ipv6 access-group common  C_acl2  ACL1 in
end
```

Note When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.

Note If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how to remove the interface ACL:

```
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl1 ACL1 in
end
```

Additional References for IPv6 ACL Chaining with a Common ACL

Related Documents

Related Topic	Document Title
IPv4 ACL Chaining Support	Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

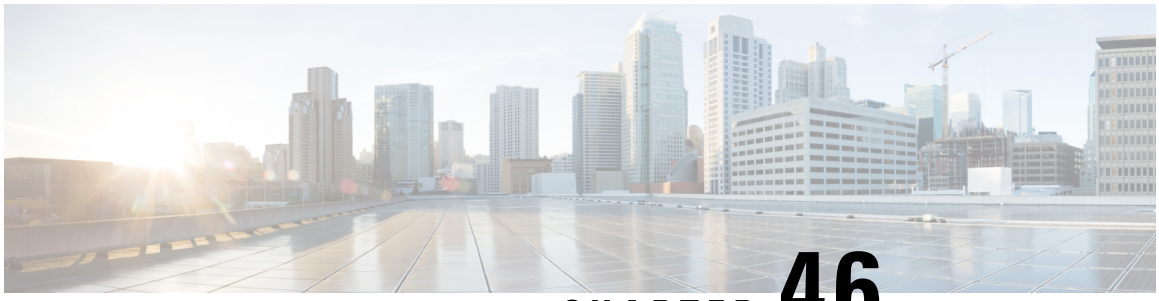
Feature Information for IPv6 ACL Chaining with a Common ACL

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 67: Feature Information for IPv6 ACL Chaining with a Common ACL

Feature Name	Releases	Feature Information
IPv6 ACL Chaining with a Common ACL	Cisco IOS XE Release 3.11S Cisco IOS XE Release 3.6E	The ACL Chaining feature, also known as Multi-ACLs, allows you to explicitly split IPv6 traffic filter access control lists (ACLs) into common and per-session ACLs. In this way, the common access control entries (ACEs) that are used reduces resource usage of each ACL entry per session in the Ternary Content Addressable Memory (TCAM). The following commands were introduced or modified: **ip access-group common**.



CHAPTER 46

IPv6 ACL Extensions for Hop by Hop Filtering

The IPv6 ACL Extensions for Hop by Hop Filtering feature allows you to control IPv6 traffic that might contain hop-by-hop extension headers. You can configure an access control list (ACL) to deny all hop-by-hop traffic or to selectively permit traffic based on protocol.

Information About IPv6 ACL Extensions for Hop by Hop Filtering

ACLs and Traffic Forwarding

IPv6 access control lists (ACLs) determine what traffic is blocked and what traffic is forwarded at device interfaces. ACLs allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Use the **ipv6 access-list** command to define an IPv6 ACL, and the **deny** and **permit** commands to configure its conditions.

The IPv6 ACL Extensions for Hop by Hop Filtering feature implements RFC 2460 to support traffic filtering in any upper-layer protocol type.

How to Configure IPv6 ACL Extensions for Hop by Hop Filtering

Configuring IPv6 ACL Extensions for Hop by Hop Filtering

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*

4. **permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
5. **deny** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list hbh-acl`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
Step 4	**permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] **Example:** `Device(config-ipv6-acl)# permit icmp any any dest-option-type`	Sets permit conditions for the IPv6 ACL.
Step 5	**deny** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*]	Sets deny conditions for the IPv6 ACL.

	Command or Action	Purpose
	[**sequence** *value*] [**time-range** *name*] [**undetermined-transport**] **Example:** `Device(config-ipv6-acl)# deny icmp any any dest-option-type`	
Step 6	**end** **Example:** `Device (config-ipv6-acl)# end`	Returns to privileged EXEC configuration mode.

Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering

Example: IPv6 ACL Extensions for Hop by Hop Filtering

```
Device(config)# ipv6 access-list hbh_acl
Device(config-ipv6-acl)# permit tcp any any hbh
Device(config-ipv6-acl)# permit tcp any any
Device(config-ipv6-acl)# permit udp any any
Device(config-ipv6-acl)# permit udp any any hbh
Device(config-ipv6-acl)# permit hbh any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# hardware statistics
Device(config-ipv6-acl)# exit

! Assign an IP address and add the ACL on the interface.

Device(config)# interface FastEthernet3/1
Device(config-if)# ipv6 address 1001::1/64
Device(config-if)# ipv6 traffic-filter hbh_acl in
Device(config-if)# exit
Device(config)# exit
Device# clear counters
Clear "show interface" counters on all interfaces [confirm]
Device#

! Verify the configurations.

Device# show running-config interface FastEthernet3/1

Building configuration...

Current configuration : 114 bytes
!
interface FastEthernet3/1
no switchport
ipv6 address 1001::1/64
ipv6 traffic-filter hbh_acl
end
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

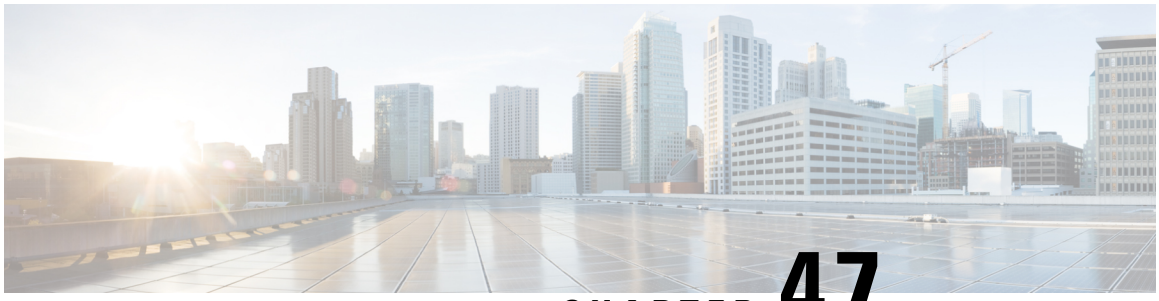
Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 68: Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

Feature Name	Releases	Feature Information
IPv6 ACL Extensions for Hop by Hop Filtering	Cisco IOS Release XE 3.4S Cisco IOS Release XE 3.5S Cisco IOS Release XE 3.6S Cisco IOS Release XE 3.3SG	Allows you to control IPv6 traffic that might contain hop-by-hop extension headers. The following commands were introduced or modified: **deny** (IPv6), **permit** (IPv6).



CHAPTER 47

Security (ACL) Enhancements

The Security (ACL) enhancements features provides you the option to restrict the number of ACLs or aces or both that can be configured on a box. Restricting the number of ACLs or aces on a box enables you to prevent depletion or over usage of tcam space which can adversely affect the performance of a box.

Restrictions

- The acl-ace-limit set is per ACL and is applicable to all the ACLs on the box.
- The acl-limit and acl-ace-limit are mutually exclusive to global-ace-limit. You cannot configure global-ace-limit when acl-limit and acl-ace-limt are configured and vice-versa.

  The limit that will be set cannot be less than the existing number of ACLs/aces in the box.
- The ACL-limit or acl-ace-limit or global-ace-limit set will be applicable to the ACLs/aces created internally while device booting up.
- The ACL with object group ace (ogace) expansion is not supported in this release, based on the customer requirements this can be investigated further. Each ogace is counted as one ace.
- The ACL-limit or acl-ace-limit or global-ace-limit set is applicable to all static and dynamically created ACLs except for template ACLs.
- The configurable ACL-limit or acl-ace-limit or global-ace-limit doesn't guarantee that the tcam space will never be overused or depleted. You must know the exact limit configurable that can be supported on the box from prior testing in the lab.
- The assumption is that all the ACLs configured on the box will be applied to the interface, which affects the tcam space.
- When the box reaches max ACL-limit or acl-ace-limit or global-ace-limit configurable, and if any client tries to create a dynamic ACL/aces then the request is rejected with the syslog error message. It is up to you to handle the failure accordingly.

Configuring Security (ACL) Enhancements

To configure ACL and ACE limits for V4 and V6:

```
enable
configure terminal
 access-list acl-limit 10
 access-list acl-ace-limit 12
 access-list global-ace-limit 14
 end
```

Note The acl-limit and acl-ace-limit are mutually exclusive to global-ace-limit.

Important Notes

- The max ACL limit range configurable is 1 to 2^{16}.
- The max ace limit range per ACL configurable is 1 to 2^{32}.
- The max global ace limit range configurable is 1 to 2^{32}.
- The acl-ace-limit set is applicable to all the ACLs that are already configured and will be configured.

Verifying Security (ACL) Enhancements Configuration

You can use the **show access-list acl-limit** command to display the number of ACLs and ACEs that are configured.

```
Device# show access-list acl-limit
Max ACLs configurable:       50
Number of ACLs configured:   10

Max aces/ACL configurable:   10

Max aces configurable:       100
Number of aces configured:   67
```

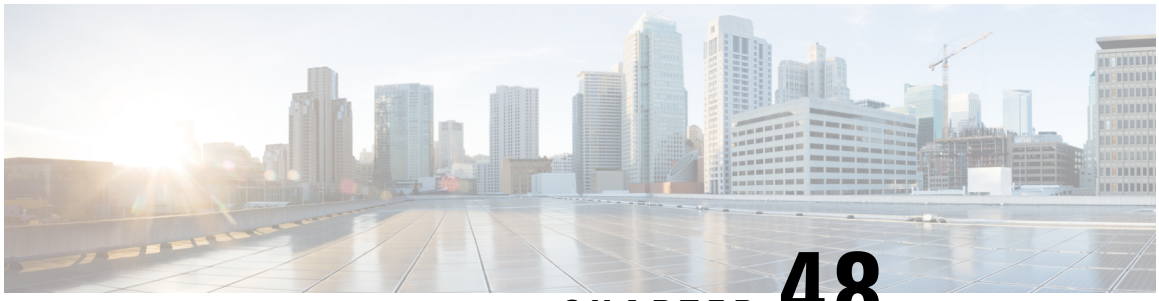
Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 69: Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

Feature Name	Releases	Feature Information
IPv6 ACL Extensions for Hop by Hop Filtering	Cisco IOS Release XE 3.4S Cisco IOS Release XE 3.5S Cisco IOS Release XE 3.6S Cisco IOS Release XE 3.3SG	Allows you to control IPv6 traffic that might contain hop-by-hop extension headers. The following commands were introduced or modified: **deny** (IPv6), **permit** (IPv6).



CHAPTER 48

IPv6 Object Groups for ACLs

The IPv6 Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply those groups to access control lists (ACLs) to create access control policies for those groups. This feature lets you use object groups instead of individual IP addresses, protocols, and ports, which are used in conventional ACLs. This feature allows multiple access control entries (ACEs), but now you can use each ACE to allow an entire group of users to access a group of servers or services or to deny them from doing so.

In large networks, the number of ACLs can be large (hundreds of lines) and difficult to configure and manage, especially if the ACLs frequently change. Object group-based ACLs are smaller, more readable, and easier to configure and manage than conventional ACLs, simplifying static and dynamic ACL deployments for large user access environments on Cisco IOS routers.

Cisco IOS Firewall benefits from object groups, because they simplify policy creation (for example, group A has access to group A services).

Restrictions for IPv6 Object Groups for ACLs

- Object group-based ACLs support only Layer 3 interfaces (such as routed interfaces and VLAN interfaces). Object group-based ACLs do not support Layer 2 features such as VLAN ACLs (VACLs) or port ACLs (PACLs).
- Object group-based ACLs are not supported with IPsec.
- The highest number of object group-based ACEs supported in an ACL is 2048.
- Empty object groups are automatically deleted.
- The object-group needs to be created before referencing it in the accesslist. An object-group cannot be deleted when it is referenced by other features, like access lists.
- Object groups that contain ACL entries are skipped, if an ACL match is performed for a packet flow.

Information About IPv6 Object Groups for ACLs

You can configure conventional Access Contro list Entries (ACEs) and ACEs that refer to object groups in the same ACL.

You can use object group-based ACLs with quality of service (QoS) match criteria, Cisco IOS Firewall, Dynamic Host Configuration Protocol (DHCP), and any other features that use extended ACLs. In addition, you can use object group-based ACLs with multicast traffic.

In larger configurations, this feature reduces the storage needed in NVRAM, because when you use object groups in ACEs, you do not need to define an individual ACE for every address and protocol pairing.

Object Groups

An object group can contain a single object (such as a single IP address, network, or subnet) or multiple objects (such as a combination of multiple IP addresses, networks, or subnets).

A typical access control entry (ACE) allows a group of users to have access only to a specific group of servers. In an object group-based access control list (ACL), you can create a single ACE that uses an object group name instead of creating many ACEs (which requires each ACE to have a different IP address). A similar object group (such as a protocol port group) can be extended to provide access only to a set of applications for a user group. ACEs can have object groups for the source only, destination only, none, or both.

You can use object groups to separate the ownership of the components of an ACE. For example, each department in an organization controls its group membership, and the administrator owns the ACE itself to control which departments can contact one another.

IPv6 addresses and services (protocols) are treated as objects, which are then grouped into various object-groups as required. The two types of object groups are v6-network (for addresses) and v6-service (for protocols) object groups. You can do the nesting of object groups if required.

The object groups can be referenced in the place of protocol or source or destination address while configuring an IPv6 ACE. The ACE containing object group(s) is expanded into individual ACEs (for each object) and programmed into the hardware.

IPv6 network and service object-groups have their own config sub-modes in which the objects are added.

You can use object groups in features that use Cisco Policy Language (CPL) class maps.

This feature supports two types of object groups for grouping ACL parameters: network object groups and service object groups. Use these object groups to group IP addresses, protocols, protocol services (ports), and Internet Control Message Protocol (ICMP) types.

Objects Allowed in Network Object Groups

A network object group is a group of any of the following objects:

- IPv6 address
- Host IPv6 addresses
- Other network object groups
- Subnets

Objects Allowed in Service Object Groups

A service object group is a group of any of the following objects:

- Source and destination protocol ports (such as Telnet or Simple Network Management Protocol [SNMP])
- Internet Control Message Protocol (ICMP) types (such as echo, echo-reply, or unreachables)
- Top-level protocols (such as Encapsulating Security Payload [ESP], TCP, or UDP)
- Other service object groups

ACLs Based on Object Groups

All features that use or reference conventional access control lists (ACLs) are compatible with object-group-based ACLs, and the feature interactions for conventional ACLs are the same with object-group-based ACLs. This feature extends the conventional ACLs to support object-group-based ACLs and also adds new keywords and the source and destination addresses and ports.

You can add, delete, or change objects in an object group membership list dynamically (without deleting and redefining the object group). Also, you can add, delete, or change objects in an object group membership list without redefining the ACL access control entry (ACE) that uses the object group. You can add objects to groups, delete them from groups, and then ensure that changes are correctly functioning within the object-group-based ACL without reapplying the ACL to the interface.

You can configure an object-group-based ACL multiple times with a source group only, a destination group only, or both source and destination groups.

You cannot delete an object group that is used within an ACL or a class-based policy language (CPL) policy.

How to Configure Object Groups for ACLs

To configure object groups for ACLs, you first create one or more object groups. These can be any combination of network object groups (groups that contain objects such as, host addresses and network addresses) or service object groups (which use operators such as **lt**, **eq**, **gt**, **neq**, and **range** with port numbers). Then, you create access control entries (ACEs) that apply a policy (such as **permit** or **deny**) to those object groups.

Configuring IPv6 Object Groups

Object Groups

The following object-groups are added:

```
Device# enable
Device# configure terminal
Device(config)# object-group ?
network     network group
security    security group
service     service group
v6-network  IPv6 network group
v6-service  IPv6 service group
```

Using Object Groups in IPv6 ACL

Object groups can be used in access-lists in 3 positions: protocol, source and destination IPv6 addresses

The following object-group options are added to existing protocol/address options.

```
Device(config-v6network-group)#?

Device(config-ipv6-acl)# [no] { permit | deny } [ <protocol options> | object-group
<v6service og name> ] { <source address options> | object-group <v6network OG
name> } { <destination address options> | object-group <v6network OG name> }
```

Creating an IPv6 Network Object Group

A network object group that contains a single object (such as a single IP address, a hostname, another network object group, or a subnet) or multiple objects with a network object-group-based ACL to create access control policies for the objects.

Perform the following steps to create IPv6 network object groups:

```
Device> enable
Device# configure terminal
Device(config)# object-group v6-network name
Device(config-v6network-group)# [no] { description <desc> | <x.x.x.x::x/prefix_len> |
host <x.x.x.x::x> | group-object <nested OG name> }


Device(config)#object-group v6-net ognet1
Device(config-v6network-group)#?


V6-Network object group configuration commands:
X:X:X:X::X/<0-128>  - IPv6 network address/prefix length
description         - Network object group description
exit                - Exit from object group configuration mode
group-object        - Nested object group
host                - Host address of group member
no                  - Negate or set default values of a command
```

Creating IPv6 Service Object Groups

Use a service object group to specify TCP and/or UDP ports or port ranges. When the service object group is associated with an access control list (ACL), this service object-group-based ACL can control access to ports.

Perform the following steps to create IPv6 service object group:

```
Device> enable
Device# configure terminal
Device(config)# object-group v6-service <name>
Device(config-v6service-group)# [no] {description <desc> | <0-255> | ahp | esp | hbh |icmp
 [<message type>]
| ipv6 | pcp | { <sctp | tcp | udp | tcp-udp> [source <src port options>]}
[<dest port options>] | group-object <nested OG name> }
Device(config-service-group)# end
```

```
Device# (config-v6service-group)#?
IPv6 Service object group configuration commands:
<0-255>         -  An IP protocol number
ahp             - Authentication Header Protocol
description     - Service object group description
```

```
esp             - Encapsulation Security Payload
exit            - Exit from object-group configuration mode
group-object    - Nested object group
hbh             - Hop by Hop options header
icmp            - Internet Control Message Protocol
ipv6            - Any Internet Protocol (v6)
no              - Negate or set default values of a command
pcp             - Payload Compression Protocol
sctp            - Streams Control Transmission Protocol
tcp             - Transmission Control Protocol
tcp-udp         - TCP or UDP protocol
udp             - User Datagram Protocol
```

Verifying IPv6 Object Groups for ACLs

Perform the following steps to verfiy IPv6 object groups for ACLs:

```
Device# enable
Device# show running int <name>-------to check if ACL is applied on the interface
Device# show object-group object-group-name --------to check if configured object groups
are referenced
Device# show ipv6 access-list --------to check the configured ACL
```

The above mentioned show commands display the contents of the named or numbered access list or object group-based ACL (or for all access lists and object group-based ACLs if no name is entered).

Configuration Examples for Object Groups for ACLs

Example: Creating an IPv6 Network Object Group

The following example shows how to create an IPv6 network object group named v6-network ognet1:

```
Device> enable
Device# configure terminal
Device(config)# object-group v6-network ognet1
Device(config-v6-network-group)# 1:1:2::0/32
Device(config-v6-network-group)# host AB:233::23D5
Device(config-v6-network-group)# exit
```

The following example shows how to create a network object group named v6-network ognet2, which contains a host, a subnet, and an existing object group (child) as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group network v6-network ognet2
Device(config-v6network-group)# 1:2:3::4/36
Device(config-v6network-group)# host AABB::CCDD
Device(config-v6network-group)# group-object ognet1
Device(config-v6network-group)# exit
```

Example: Creating a IPv6 Service Object Group

The following example shows how to create a service object group named v6-service ogserv1, which contains several ICMP, TCP, UDP, and TCP-UDP protocols as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group service v6-service ogserv1
Device(config-v6service-group)# icmp unreachable
Device(config-v6service-group)# tcp smtp
Device(config-v6service-group)# tcp telnet
Device(config-v6service-group)# tcp source range 3000 4000 telnet
Device(config-v6service-group)# pcp
Device(config-v6service-group)# udp domain
Device(config-v6service-group)# hph
Device(config-v6service-group)# exit
```

Example: Creating an IPv6 Object Group-Based ACL

The following example shows how to create an IPv6 object-group-based ACL that permits packets:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list ogacl1
Device(config-ipv6-acl)# permit object-group ogserv1 5:6:7::5/56 object-group ognet1
Device(config-ipv6-acl)# deny ip object-group ognet2 object-group ognet3
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
```

Example: Verifying IPv6 Object Groups for ACLs

The following example shows how to display all object groups:

```
Device# show object-group

V6-Network object group ognet1
1:1:2::/32
host AB:233::23D5
V6-Network object group ognet2
1:2:3::4/36
host AABB::CCDD
group-object ognet1
V6-Network object group ognet3
host 1::1
host 1::2
host 1::3
V6-Service object group ogserv1
icmp unreachable
tcp source range 3000 4000 eq telnet
pcp
hbh
```

The following example shows how to display information about IPv6 object-group-based ACL:

```
Device# show ipv6 access-list
IPv6 access list ogacl1
      permit object-group ogserv1 5:6:7::/56 object-group ognet1 sequence 10
      deny ipv6 object-group ognet2 object-group ognet3 sequence 20
      permit ipv6 any any sequence 30
```

Additional References for Object Groups for ACLs

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
ACL configuration guide	*Security Configuration Guide: Access Control Lists*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Object Groups for ACLs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

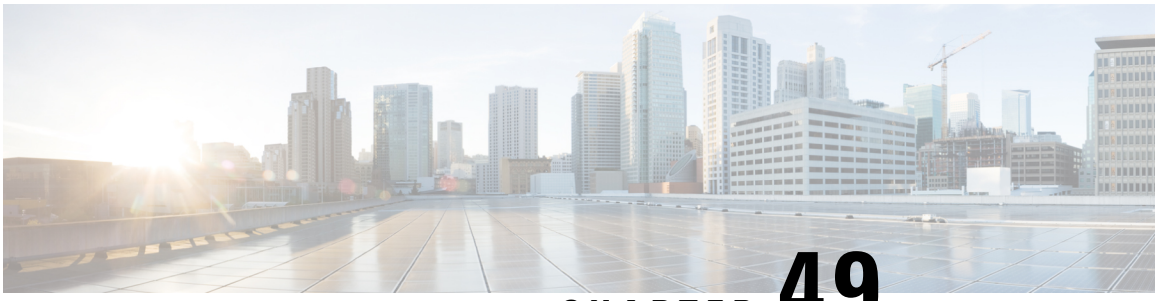
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 70: Feature Information for Object Groups for ACLs

Feature Name	Releases	Feature Information
IPv6 Object Groups for ACLs	Cisco IOS XE Release 16.11.1	The IPv6 Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply them to access control lists (ACLs) to create access control policies for those groups. This feature lets you use object groups instead of individual IP addresses, protocols, and ports, which are used in conventional ACLs. This feature allows multiple access control entries (ACEs), but now you can use each ACE to allow an entire group of users to access a group of servers or services or to deny them from doing so.

PART IV

RADIUS



CHAPTER 49

Configuring RADIUS

The RADIUS security system is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

Prerequisites for RADIUS

To configure RADIUS on your Cisco device or access server, you must perform these tasks:

- Use the **aaa new-model** global configuration command to enable Authentication, Authorization, and Accounting (AAA). AAA must be configured if you plan to use RADIUS.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication.
- Use **line** and **interface** commands to enable the defined method lists to be used.

Restrictions for RadSec (RADIUS Security)

RadSec is not supported on any of the Cisco enterprise routing platforms.

Information About RADIUS

RADIUS Network Environments

Cisco supports RADIUS under its authentication, authorization, and accounting (AAA) security paradigm. RADIUS can be used with other AAA security protocols such as TACACS+, Kerberos, and local username lookup. RADIUS is supported on all Cisco platforms, but some RADIUS-supported features run only on specified platforms.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a smart card access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco device with RADIUS to the network. This might be the first step when you make a transition to a TACACS+ server.
- Networks in which a user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to a single protocol such as PPP. For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using the IP address 10.2.3.4 and the defined access list is started.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, and bytes) used during the session. An ISP might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions, and to efficiently manage the use of shared resources to offer differing service-level agreements.

RADIUS is not suitable in these network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
  - AppleTalk Remote Access (ARA)
  - NetBIOS Frame Control Protocol (NBFCP)
  - NetWare Asynchronous Services Interface (NASI)
  - X.25 Packet Assemblers/Disassemblers (PAD) connections

- Device-to-device situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

1. The user is prompted to enter the username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   a. ACCEPT—The user is authenticated.
   b. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   c. CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.
   d. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including connections such as Telnet, rlogin, or local-area transport (LAT), and services such as PPP, Serial Line Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

RADIUS Attributes

The network access server monitors the RADIUS authorization and accounting functions defined by RADIUS attributes in each user profile:

Vendor-Proprietary RADIUS Attributes

An Internet Engineering Task Force (IETF) standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. Some vendors, nevertheless, have extended the RADIUS attribute set in a unique way. Cisco software supports a subset of vendor-proprietary RADIUS attributes.

RADIUS Tunnel Attributes

RADIUS is a security server AAA protocol originally developed by Livingston, Inc. RADIUS uses attribute value (AV) pairs to communicate information between the security server and the network access server.

RFC 2138 and RFC 2139 describe the basic functionality of RADIUS and the original set of IETF-standard AV pairs used to send AAA information. Two IETF standards, “RADIUS Attributes for Tunnel Protocol Support” and “RADIUS Accounting Modifications for Tunnel Protocol Support,” extend the IETF-defined

set of AV pairs to include attributes specific to VPNs. These attributes are used to carry the tunneling information between the RADIUS server and the tunnel initiator.

RFC 2865 and RFC 2868 extend the IETF-defined set of AV pairs to include attributes specific to compulsory tunneling in VPNs by allowing the user to specify authentication names for the network access server and the RADIUS server.

Cisco devices and access servers support new RADIUS IETF-standard virtual private dialup network (VPDN) tunnel attributes.

Preauthentication on a RADIUS Server

RADIUS attributes are configured in the RADIUS preauthentication profiles to specify preauthentication behavior. In addition to configuring preauthentication on your Cisco device, you must set up the preauthentication profiles on the RADIUS server.

RADIUS Profile for DNIS or CLID Preauthentication

To configure the RADIUS preauthentication profile, use the Dialed Number Identification Service (DNIS) or Calling Line Identification (CLID) number as the username, and use the password defined in the **dnis** or **clid** command as the password.

Note The preauthentication profile must have "outbound" as the service type because the password is predefined on the network access server (NAS). Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The "outbound" service type is also included in the Access-Request packet sent to the RADIUS server.

RADIUS Profile for Call Type Preauthentication

To set up the RADIUS preauthentication profile, use the call type string as the username, and use the password defined in the **ctype** command as the password. The table below lists the call type strings that can be used in the preauthentication profile.

Table 71: Call Type Strings Used in Preauthentication

Call Type String	ISDN Bearer Capabilities
digital	Unrestricted digital, restricted digital.
speech	Speech, 3.1 kHz audio, 7 kHz audio. **Note** This is the only call type available for channel-associated signaling (CAS) .
v.110	Anything with the V.110 user information layer.
v.120	Anything with the V.120 user information layer.

Note The preauthentication profile must have “outbound” as the service type because the password is predefined on the NAS. Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The “outbound” service type is also included in the Access-Request packet sent to the RADIUS server and should be a checkin item if the RADIUS server supports checkin items.

RADIUS Profile for Preauthentication Enhancements for Callback

Callback allows remote network users such as telecommuters to dial in to the NAS without being charged. When callback is required, the NAS hangs up the current call and dials the caller back. When the NAS performs the callback, only information for the outgoing connection is applied. The rest of the attributes from the preauthentication access-accept message are discarded.

Note The destination IP address is not required to be returned from the RADIUS server.

The following example shows a RADIUS profile configuration with a callback number of 555-0101 and the service type set to outbound. The cisco-avpair = “preauth:send-name=<string>” uses the string “user1” and the cisco-avpair = “preauth:send-secret=<string>” uses the password “cisco.”

```
5550101 password = “cisco”, Service-Type = Outbound
Service-Type = Callback-Framed
Framed-Protocol = PPP,
Dialback-No = “5550119”
Class = “ISP12”
cisco-avpair = “preauth:send-name=user1”
cisco-avpair = “preauth:send-secret=cisco”
```

RADIUS Profile for a Remote Hostname Used for Large-Scale Dial-Out

The following example protects against accidentally calling a valid telephone number but accessing the wrong device by providing the name of the remote device, for use in large-scale dial-out:

```
5550101 password = "PASSWORD1", Service-Type = Outbound
       Service-Type = Callback-Framed
       Framed-Protocol = PPP,
       Dialback-No = "5550190"
       Class = "ISP12"
       cisco-avpair = "preauth:send-name=user1"
       cisco-avpair = "preauth:send-secret=PASSWORD1"
       cisco-avpair = "preauth:remote-name=Device2"
```

RADIUS Profile for Modem Management

When DNIS, CLID, or call type preauthentication is used, the affirmative response from the RADIUS server might include a modem string for modem management in the NAS through vendor-specific attribute (VSA) 26. The modem management VSA has this syntax:

```
cisco-avpair = “preauth:modem-service=modem min-speed <
x
> max-speed <
```

```
y
>
modulation <
z
> error-correction <
a
> compression <
b
>”
```

The table below lists the modem management string elements within the VSA.

Table 72: Modem Management String

Command	Argument
min-speed	300 to 56000, any
max-speed	300 to 56000, any
modulation	K56Flex, v22bis, v32bis, v34, v90, any
error-correction	lapm, mnp4
compression	mnp5, v42bis

When the modem management string is received from the RADIUS server in the form of a VSA, the information is passed to the Cisco software and applied on a per-call basis. Modem ISDN channel aggregation (MICA) modems provide a control channel through which messages can be sent during the call setup time. Hence, this modem management feature is supported only with MICA modems. This feature is not supported with Microcom modems.

RADIUS Profile for Subsequent Authentication

If preauthentication passes, you can use vendor-proprietary RADIUS attribute 201 (Require-Auth) in the preauthentication profile to determine whether subsequent authentication is performed. If attribute 201, returned in the access-accept message, has a value of 0, subsequent authentication is not performed. If attribute 201 has a value of 1, subsequent authentication is performed as usual.

Attribute 201 has this syntax:

```
cisco-avpair = “preauth:auth-required=<
n
>”
```

where <*n*> has the same value range as attribute 201 (that is, 0 or 1).

If attribute 201 is missing in the preauthentication profile, a value of 1 is assumed, and subsequent authentication is performed.

Note Before you can perform subsequent authentication, you must set up a regular user profile in addition to a preauthentication profile.

RADIUS Profile for Subsequent Authentication Types

If you specified subsequent authentication in the preauthentication profile, you must also specify the authentication types to be used for subsequent authentication. To specify the authentication types allowed in subsequent authentication, use this VSA:

```
cisco-avpair = "preauth:auth-type=<
string
>"
```

The table below lists the allowed values for the <*string*> element.

Table 73: <string> Element Values

String	Description
chap	Requires the username and password for the Challenge-Handshake Authentication Protocol (CHAP) for PPP authentication.
ms-chap	Requires the username and password for the MS-CHAP for PPP authentication.
pap	Requires the username and password for the Password Authentication Protocol (PAP) for PPP authentication.

To specify that multiple authentication types are allowed, you can configure more than one instance of this VSA in the preauthentication profile. The sequence of the authentication type VSAs in the preauthentication profile is significant because it specifies the order of authentication types to be used in the PPP negotiation.

This VSA is a per-user attribute and replaces the authentication type list in the **ppp authentication** interface configuration command.

Note You should use this VSA only if subsequent authentication is required because it specifies the authentication type for subsequent authentication.

RADIUS Profile to Include the Username

If only preauthentication is used to authenticate a call, the NAS could be missing a username when it brings up the call. RADIUS can provide a username for the NAS to use through RADIUS attribute 1 (User-Name) or through a VSA returned in the Access-Accept packet. The VSA for specifying the username has this syntax:

```
cisco-avpair = "preauth:username=<
string
>"
```

If no username is specified, the DNIS number, CLID number, or call type is used, depending on the last preauthentication command configured (for example, if **clid** was the last preauthentication command configured, the CLID number is used as the username).

If subsequent authentication is used to authenticate a call, there might be two usernames: one provided by RADIUS and one provided by the user. In this case, the username provided by the user overrides the one contained in the RADIUS preauthentication profile. The username provided by the user is used for both authentication and accounting.

RADIUS Profile for Two-Way Authentication

In the case of two-way authentication, the calling networking device must authenticate the NAS. The PAP username and password or CHAP username and password need not be configured locally on the NAS. Instead, the username and password can be included in the Access-Accept messages for preauthentication.

Note Do not configure the **ppp authentication** command with the **radius** command.

To set up PAP, do not configure the **ppp pap sent-name password** command on the interface. The VSAs "preauth:send-name" and "preauth:send-secret" are used as the PAP username and PAP password for outbound authentication.

For CHAP, "preauth:send-name" is used not only for outbound authentication but also for inbound authentication. For a CHAP inbound case, the NAS uses the name defined in "preauth:send-name" in the challenge packet to the caller networking device. For a CHAP outbound case, both "preauth:send-name" and "preauth:send-secret" are used in the response packet.

The following example shows a configuration that specifies two-way authentication:

```
5550101 password = "PASSWORD2", Service-Type = Outbound
Service-Type = Framed-User
cisco-avpair = "preauth:auth-required=1"
cisco-avpair = "preauth:auth-type=pap"
cisco-avpair = "preauth:send-name=user1"
cisco-avpair = "preauth:send-secret=PASSWORD2"
class = "<some class>"
```

Note Two-way authentication does not work when resource pooling is enabled.

RADIUS Profile to Support Authorization

If only preauthentication is configured, subsequent authentication is bypassed. Note that because the username and password are not available, authorization is also bypassed. However, you can include authorization attributes in the preauthentication profile to apply per-user attributes and avoid having to return subsequently to RADIUS for authorization. To initiate the authorization process, you must also configure the **aaa authorization network** command on the NAS.

You can configure authorization attributes in the preauthentication profile with one exception: the service-type attribute (attribute 6). The service-type attribute must be converted to a VSA in the preauthentication profile. This VSA has this syntax:

```
cisco-avpair = "preauth:service-type=<
n
>"
```

where <*n*> is one of the standard RFC 2865 values for attribute 6.

Note If subsequent authentication is required, the authorization attributes in the preauthentication profile are not applied.

RADIUS Authentication

After you have identified the RADIUS server and defined the RADIUS authentication key, you must define method lists for RADIUS authentication. Because RADIUS authentication is facilitated through AAA, you must enter the **aaa authentication** command, specifying RADIUS as the authentication method.

RADIUS Authorization

AAA authorization lets you set parameters that restrict a user's access to the network. Authorization using RADIUS provides one method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, AppleTalk Remote Access (ARA), and Telnet. Because RADIUS authorization is facilitated through AAA, you must enter the **aaa authorization** command, specifying RADIUS as the authorization method.

RADIUS Accounting

The AAA accounting feature enables you to track the services users are accessing and the amount of network resources they are consuming. Because RADIUS accounting is facilitated through AAA, you must enter the **aaa accounting** command, specifying RADIUS as the accounting method.

RADIUS Login-IP-Host

To enable the network access server (NAS) to attempt more than one login host when trying to connect a dial-in user, you can enter as many as three Login-IP-Host entries in the user's profile on the RADIUS server. The following example shows that three Login-IP-Host instances are configured for the user *user1*, and that TCP-Clear is used for the connection:

```
user1 Password = xyz
        Service-Type = Login,
        Login-Service = TCP-Clear,
        Login-IP-Host = 10.0.0.0,
        Login-IP-Host = 10.2.2.2,
        Login-IP-Host = 10.255.255.255,
        Login-TCP-Port = 23
```

The order in which the hosts are entered is the order in which they are attempted. Use the **ip tcp synwait-time** command to set the number of seconds that the NAS waits before trying to connect to the next host on the list; the default is 30 seconds.

Your RADIUS server might permit more than three Login-IP-Host entries; however, the NAS supports only three hosts in Access-Accept packets.

RADIUS Prompt

To control whether user responses to Access-Challenge packets are echoed to the screen, you can configure the Prompt attribute in the user profile on the RADIUS server. This attribute is included only in Access-Challenge packets. The following example shows the Prompt attribute set to No-Echo, which prevents the user's responses from echoing:

```
user1 Password = xyz
Service-Type = Login,
```

```
Login-Service = Telnet,
Prompt = No-Echo,
Login-IP-Host = 172.31.255.255
```

To allow user responses to echo, set the attribute to Echo. If the Prompt attribute is not included in the user profile, responses are echoed by default.

This attribute overrides the behavior of the **radius-server challenge-noecho** command configured on the access server. For example, if the access server is configured to suppress echoing, but the individual user profile allows echoing, the user responses are echoed.

Note If you want to use the Prompt attribute, your RADIUS server must be configured to support Access-Challenge packets.

Vendor-Specific RADIUS Attributes

The IETF standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor ID is 9, and the supported option has vendor type 1, which is named "cisco-avpair." The value is a string with this format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, Internetwork Packet Exchange (IPX), VPDN, VoIP, Secure Shell (SSH), Resource Reservation Protocol (RSVP), Serial Interface Processor (SIP), AirNet, and Outbound. "Attribute" and "value" are an appropriate AV pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes, allowing the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's Internet Protocol Control Protocol (IPCP) address assignment):

```
cisco-avpair= ”ip:addr-pool=first“
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional:

```
cisco-avpair= ”ip:addr-pool*first“
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= ”shell:priv-lvl=15“
```

Other vendors have their own unique vendor IDs, options, and associated VSAs.

Static Routes and IP Addresses on the RADIUS Server

Some vendor-proprietary implementations of RADIUS let the user define static routes and IP pool definitions on the RADIUS server instead of on each individual network access server in the network. Each network access server then queries the RADIUS server for static route and IP pool information.

To have the Cisco device or access server query the RADIUS server for static routes and IP pool definitions when the device starts up, use the **radius-server configure-nas** command.

Because the **radius-server configure-nas** command is performed when the Cisco devcie starts up, it does not take effect until you enter a **copy system:running-config nvram:startup-config** command.

How to Configure RADIUS

Configuring a Device for Vendor-Proprietary RADIUS Server Communication

Although an IETF standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco software supports a subset of vendor-proprietary RADIUS attributes.

To configure RADIUS (whether vendor-proprietary or IETF compliant), you must use the **radius-server** commands to specify the host running the RADIUS server daemon and the secret text string it shares with the Cisco device. To identify that the RADIUS server is using a vendor-proprietary implementation of RADIUS, use the **radius-server host non-standard** command. Vendor-proprietary attributes are not supported unless you use the **radius-server host non-standard** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send** [**accounting** | **authentication**]
4. **radius server** *server-name*
5. **address ipv4** *ip-address*
6. **non-standard**
7. **key** {**0** *string* | **7** *string* | *string*}
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**radius-server vsa send** [**accounting** | **authentication**] **Example:** `Device(config)# radius-server vsa send`	Enables the network access server to recognize and use VSAs as defined by RADIUS IETF attribute 26.
Step 4	**radius server** *server-name* **Example:** `Device(config)# radius server rad1`	Specifies the name for the RADIUS server. **Note** The **radius-server host** command is deprecated from Cisco IOS Release 15.4(2)S. To configure an IPv4 or IPv6 RADIUS server, use the **radius server** *name* command. For more information about the **radius server** command, see *Cisco IOS Security Command Reference: Commands M to R*.
Step 5	**address ipv4** *ip-address* **Example:** `Device(config-radius-server)# address ipv4 10.45.1.2`	Assigns an IP address to the RADIUS server.
Step 6	**non-standard** **Example:** `Device(config-radius-server)# non-standard`	Identifies that the security server is using a vendor-proprietary implementation of RADIUS.
Step 7	**key** {**0** *string* | **7** *string* | *string*} **Example:** `Device(config-radius-server)# key myRaDIUSpassword`	Specifies the shared secret text string used between the device and the vendor-proprietary RADIUS server. • The device and the RADIUS server use this text string to encrypt passwords and exchange responses.
Step 8	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Configuring a Device to Expand Network Access Server Port Information

Sometimes PPP or login authentication occurs on an interface that is different from the interface on which the call itself comes in. For example, in a V.120 ISDN call, login or PPP authentication occurs on a virtual asynchronous interface “ttt”, but the call itself occurs on one of the channels of the ISDN interface.

The **radius-server attribute nas-port extended** command configures RADIUS to expand the size of the NAS-Port attribute (RADIUS IETF attribute 5) field to 32 bits. The upper 16 bits of the NAS-Port attribute display the type and number of the controlling interface; the lower 16 bits indicate the interface undergoing authentication.

Note The **radius-server attribute nas-port format** command replaces the **radius-server extended-portnames** command and the **radius-server attribute nas-port extended** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server configure-nas**
4. **radius-server attribute nas-port format**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server configure-nas** **Example:** `Device(config)# radius-server configure-nas`	(Optional) Tells the Cisco device or access server to query the RADIUS server for the static routes and IP pool definitions used throughout its domain. **Note** Because the **radius-server configure-nas** command is used when the Cisco device starts up, it does not take effect until you issue a **copy system:running-config nvram:startup-config** command.
Step 4	**radius-server attribute nas-port format** **Example:** `Device(config)# radius-server attribute nas-port format`	Expands the size of the NAS-Port attribute from 16 to 32 bits to display extended interface information.
Step 5	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Replacing the NAS-Port Attribute with the RADIUS Attribute

On platforms with multiple interfaces (ports) per slot, the Cisco RADIUS implementation does not provide a unique NAS-Port attribute that permits distinguishing between the interfaces. For example, if a dual PRI is in slot 1, calls on both Serial1/0:1 and Serial1/1:1 appear as NAS-Port = 20101 because of the 16-bit field size limitation associated with the RADIUS IETF NAS-Port attribute. In this case, you can replace the NAS-Port attribute with a VSA (RADIUS IETF attribute 26). Cisco's vendor ID is 9, and the Cisco-NAS-Port attribute is subtype 2. VSAs can be turned on by entering the **radius-server vsa send** command. The port information in this attribute is provided and configured using the **aaa nas port extended** command.

The standard NAS-Port attribute (RADIUS IETF attribute 5) is sent. If you do not want this information to be sent, you can suppress it by using the **no radius-server attribute nas-port** command. After this command is configured, the standard NAS-Port attribute is no longer sent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send** [**accounting** | **authentication**]
4. **aaa nas port extended**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server vsa send** [**accounting**	**authentication**] **Example:** `Device(config)# radius-server vsa send`	Enables the network access server to recognize and use vendor-specific attributes as defined by RADIUS IETF attribute 26.
Step 4	**aaa nas port extended** **Example:** `Device(config)# aaa nas port extended`	Expands the size of the VSA NAS-Port field from 16 to 32 bits to display extended interface information.
Step 5	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Monitoring and Maintaining RADIUS

SUMMARY STEPS

1. **enable**
2. **debug radius**
3. **show radius statistics**
4. **show aaa servers**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Device# debug radius`	Displays information associated with RADIUS.
Step 3	**show radius statistics** **Example:** `Device# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets. **Note** Few IOS processes use ephemeral source ports for RADIUS, and the port numbers may vary every time.
Step 4	**show aaa servers** **Example:** `Device# show aaa servers`	Displays the status and number of packets that are sent to and received from all public and private AAA RADIUS servers as interpreted by the AAA Server MIB.
Step 5	**exit** **Example:** `Device# exit`	Exits the device session.

Configuration Examples for RADIUS

Example: RADIUS Authentication and Authorization

The following example shows how to configure the device to authenticate and authorize using RADIUS:

```
aaa authentication login use-radius group radius local
aaa authentication ppp user-radius if-needed group radius
aaa authorization exec default group radius
aaa authorization network default group radius
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login use-radius group radius local** command configures the device to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database. In this example, **use-radius** is the name of the method list, which specifies RADIUS and then local authentication.
- The **aaa authentication ppp user-radius if-needed group radius** command configures the Cisco software to use RADIUS authentication for lines using PPP with CHAP or PAP if the user has not already been authorized. If the EXEC facility has authenticated the user, RADIUS authentication is not performed. In this example, **user-radius** is the name of the method list defining RADIUS as the if-needed authentication method.
- The **aaa authorization exec default group radius** command sets the RADIUS information that is used for EXEC authorization, autocommands, and access lists.
- The **aaa authorization network default group radius** command sets RADIUS for network authorization, address assignment, and access lists.

Example: RADIUS Authentication, Authorization, and Accounting

The following example shows a general configuration using RADIUS with the AAA command set:

```
radius-server host 10.45.1.2
radius-server key myRaDiUSpassWoRd
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem ri-is-cd
interface group-async 1
 encaps ppp
 ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host** command defines the IP address of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.

- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the lines specified.

Example: Vendor-Proprietary RADIUS Configuration

The following example shows a general configuration using vendor-proprietary RADIUS with the AAA command set:

Note The **radius-server host** command is deprecated from Cisco IOS Release 15.4(2)S. To configure an IPv4 or IPv6 RADIUS server, use the **radius server** *name* command. For more information about the **radius server** command, see *Cisco IOS Security Command Reference: Commands M to R*.

```
radius server myserver
radius server address ipv4 192.0.2.2
non-standard
key 7 any key
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
```

The lines in this RADIUS authentication, authorization, and accounting configuration example are defined as follows:

- The **non-standard** command defines the name of the RADIUS server host and identifies that this RADIUS host uses a vendor-proprietary version of RADIUS.
- The **key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **configure-nas** command defines that the Cisco device or access server queries the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.
- The **aaa authorization network default group radius local** command assigns an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.

- The **aaa authentication login admins local** command defines another method list, “admins,” for login authentication.

Example: Multiple RADIUS Server Entries for the Same Server IP Address

The following example shows how to configure the network access server to recognize several RADIUS host entries with the same IP address. Two different host entries on the same RADIUS server are configured for the same services—authentication and accounting. The second host entry configured acts as failover backup to the first one. (The RADIUS host entries are tried in the order they are configured.)

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group radius
! The next set of commands configures multiple host entries for the same IP address.
radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
radius-server host 172.20.0.1 auth-port 2000 acct-port 2001
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
AAA and RADIUS commands	*Cisco IOS Security Command Reference*
RADIUS attributes	*RADIUS Attributes Configuration Guide* (part of the Securing User Services Configuration Library)
AAA	*Authentication, Authorization, and Accounting Configuration Guide* (part of the Securing User Services Configuration Library)
L2TP, VPN, or VPDN	*Dial Technologies Configuration Guide* and *VPDN Configuration Guide*
Modem configuration and management	*Dial Technologies Configuration Guide*
RADIUS port identification for PPP	*Wide-Area Networking Configuration Guide*

RFCs

RFC	Title
RFC 2138	*Remote Authentication Dial-In User Service (RADIUS)*
RFC 2139	*RADIUS Accounting*
RFC 2865	*RADIUS*

RFC	Title
RFC 2867	*RADIUS Accounting Modifications for Tunnel Protocol Support*
RFC 2868	*RADIUS Attributes for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring RADIUS

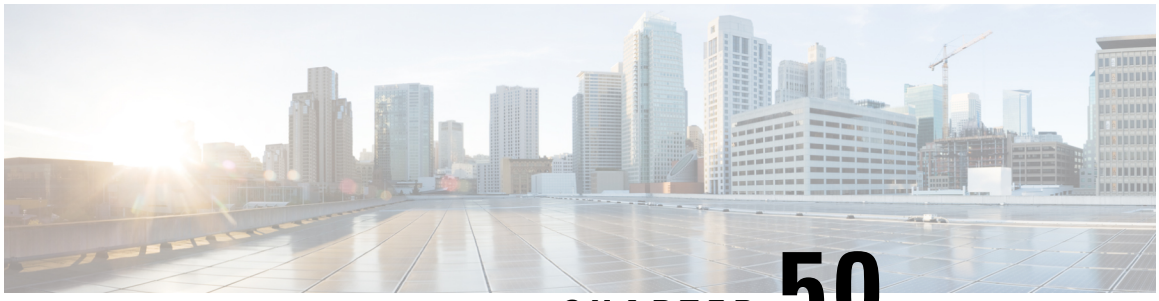
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 74: Feature Information for Configuring RADIUS

Feature Name	Releases	Feature Information
Configuring RADIUS		The RADIUS security system is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information. • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller • Catalyst 3650 Series Switches

Feature Name	Releases	Feature Information
RADIUS Statistics via SNMP		This feature provides statistics related to RADIUS traffic and private RADIUS servers. • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller • Catalyst 3650 Series Switches The following commands were introduced or modified: **show aaa servers**, **show radius statistics**.



CHAPTER 50

RADIUS for Multiple UDP Ports

RADIUS security servers are identified on the basis of their hostname or IP address, hostname and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific authentication, authorization, and accounting (AAA) service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a failover backup to the first one. If the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services.

Prerequisites for RADIUS for Multiple UDP Ports

To configure RADIUS on your Cisco device or access server, you must perform these tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use RADIUS.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication.
- Use **line** and **interface** commands to enable the defined method lists to be used.

Information About RADIUS for Multiple UDP Ports

Device-to-RADIUS Server Communication

The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (CiscoSecure ACS), Livingston, Merit, Microsoft, or another software provider. Configuring device to RADIUS server communication can have several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Timeout period
- Retransmission value
- Key string

RADIUS security servers are identified on the basis of their hostname or IP address, hostname and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a failover backup to the first one. If the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order they are configured.)

A RADIUS server and a Cisco device use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the device.

The timeout, retransmission, and encryption key values are configurable globally for all RADIUS servers, on a per-server basis or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the device, use the three unique global commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command.

Note You can configure both global and per-server timeout, retransmission, and key value commands simultaneously on the same Cisco network access server. If both global and per-server functions are configured on a device, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

How to Configure RADIUS for Multiple UDP Ports

Configuring Device-to-RADIUS Server Communication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server-name*
4. **address ipv4** *ip-address*
5. **key** {**0** *string* | **7** *string* | *string*}
6. **retransmit** *retries*
7. **timeout** *seconds*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *server-name* **Example:** `Device(config)# radius server rad1`	Specifies the name for the RADIUS server.
Step 4	**address ipv4** *ip-address* **Example:** `Device(config-radius-server)# address ipv4 10.45.1.2`	Assigns an IP address to the RADIUS server.
Step 5	**key** {**0** *string*	**7** *string*	*string*} **Example:** `Device(config-radius-server)# key myRaDIUSpassword`	Specifies the shared secret text string used between the device and a RADIUS server. **Note** In this step, the encryption key value is configured globally for all RADIUS servers.

	Command or Action	Purpose
		• Use the **0** *string* option to configure an unencrypted shared secret. Use the **7** *string* option to configure an encrypted shared secret.
Step 6	**retransmit** *retries* **Example:** `Device(config-radius-server)# retransmit 25`	Specifies how many times the device transmits each RADIUS request to the server before giving up (the default is 3). **Note** In this step, the retransmission value is configured globally for all RADIUS servers.
Step 7	**timeout** *seconds* **Example:** `Device(config-radius-server)# timeout 6`	Specifies for how many seconds a device waits for a reply to a RADIUS request before retransmitting the request. **Note** In this step, the timeout value is configured globally for all RADIUS servers.
Step 8	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Configuration Examples for RADIUS for Multiple UDP Ports

Example: Device-to-RADIUS Server Communication

The following example shows how to configure two RADIUS servers with specific timeout, retransmit, and key values. In this example, the **aaa new-model** command enables AAA services on the device, and specific AAA commands define the AAA services. The **retransmit** command changes the global retransmission value to 4 for all RADIUS servers. The **host** command configures specific timeout, retransmission, and key values for the RADIUS server hosts with IP addresses 172.16.1.1 and 172.29.39.46.

```
! Enable AAA services on the device and define those services.
aaa new-model
aaa authentication login default group radius
aaa authentication login console-login none
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
enable password tryit1
!
Device(config)# radius server rad1
Device(config-radius-server)# address ipv4 10.45.1.2
Device(config-radius-server)# key myRaDIUSpassword
Device(config-radius-server)# retransmit 25
Device(config-radius-server)# timeout 6
Device(config)# exit
```

Example: RADIUS Server with Server-Specific Values

The following example shows how to configure server-specific timeout, retransmit, and key values for the RADIUS server with IP address 172.31.39.46:

```
radius-server host 172.31.39.46 timeout 6 retransmit 5 key rad123
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
AAA	*Authentication, Authorization, and Accounting Configuration Guide* (part of the Securing User Services Configuration Library)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

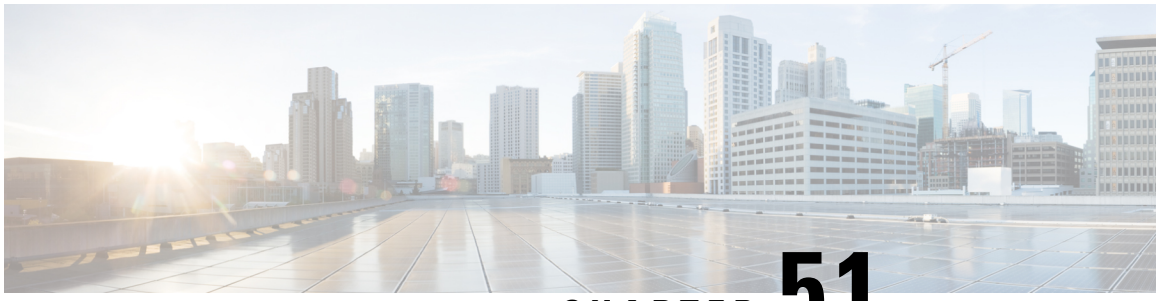
Feature Information for RADIUS for Multiple UDP Ports

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 75: Feature Information for RADIUS for Multiple UDP Ports

Feature Name	Releases	Feature Information
RADIUS for Multiple UDP Ports		RADIUS security servers are identified on the basis of their hostname or IP address, hostname and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. • Catalyst 3850 Series Switches • Catalyst 3650 Series Switches The following command was introduced or modified: **radius-server host**.



CHAPTER 51

AAA DNIS Map for Authorization

The AAA DNIS Map for Authorization feature allows you to assign a Dialed Number Identification Service (DNIS) number to a particular authentication, authorization, and accounting (AAA) server group so that the server group can process authentication, authorization, and accounting requests for users dialing in to the network using that particular DNIS. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

Prerequisites for AAA DNIS Map for Authorization

- Before configuring the device to select a particular AAA server group based on the DNIS of the server group, you must configure the list of RADIUS server hosts and AAA server groups.
- Before configuring AAA preauthentication, you must configure the **aaa new-model** command and make sure that the supporting preauthentication application is running on a RADIUS server in your network.

Information About AAA DNIS Map for Authorization

AAA Server Group Selection Based on DNIS

Cisco software allows you to assign a DNIS number to a particular AAA server group so that the server group can process authentication, authorization, and accounting requests for users dialing in to the network using that particular DNIS. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco devices with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different RADIUS server groups for different customers (that is, different RADIUS servers for different DNIS numbers). Additionally, using server groups, you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because each of these AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify or determine which server group provides AAA services, this method takes precedence over any additional AAA selection method.
- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the least precedence.

AAA Preauthentication

Configuring AAA preauthentication with ISDN PRI or channel-associated signaling (CAS) allows service providers to better manage ports using their existing RADIUS solutions and efficiently manage the use of shared resources to offer differing service-level agreements. With ISDN PRI or CAS, information about an incoming call is available to the network access server (NAS) before the call is connected. The available call information includes the following:

- The DNIS number, also referred to as the called number
- The Calling Line Identification (CLID) number, also referred to as the calling number
- The call type, also referred to as the bearer capability

The AAA preauthentication feature allows a Cisco NAS to decide--on the basis of the DNIS number, the CLID number, or the call type--whether to connect an incoming call. (With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.)

When an incoming call arrives from the public network switch, but before it is connected, AAA preauthentication enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, the NAS accepts the call. If the server does not authorize the call, the NAS sends a disconnect message to the public network switch to reject the call.

In the event that the RADIUS server application becomes unavailable or is slow to respond, a guard timer can be set in the NAS. When the timer expires, the NAS uses a configurable parameter to accept or reject the incoming call that has no authorization.

The AAA preauthentication feature supports the use of attribute 44 by the RADIUS server application and the use of RADIUS attributes that are configured in the RADIUS preauthentication profiles to specify preauthentication behavior. They can also be used, for instance, to specify whether subsequent authentication should occur and, if so, what authentication method should be used.

The following restrictions apply to AAA preauthentication with ISDN PRI and CAS:

- Attribute 44 is available for CAS calls only when preauthentication or resource pooling is enabled.
- Multichassis Multilink PPP (MMP) is not available with ISDN PRI.
- AAA preauthentication is available only on some hardware platforms.
- ISDN PRI is supported only on some hardware platforms.

Guard Timer for Call Handling

Because response times for preauthentication and authentication requests can vary, the guard timer allows you to control the handling of calls. The guard timer starts when the DNIS is sent to the RADIUS server. If the NAS does not receive a response from AAA before the guard timer expires, it accepts or rejects the calls on the basis of the configuration of the timer.

How to Configure AAA DNIS Map for Authorization

Configuring AAA DNIS Preauthentication

DNIS preauthentication enables preauthentication at call setup based on the number dialed. The DNIS number is sent directly to the security server when a call is received. If the call authenticated by AAA, it is accepted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa preauthorization**
4. **group** {**radius** | **tacacs+** | *server-group*}
5. **dnis** [**password** *string*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa preauthorization** **Example:** `Device(config)# aaa preauthorization`	Enters AAA preauthentication configuration mode.
Step 4	**group** {**radius** | **tacacs+** | *server-group*} **Example:** `Device(config-preauth)# group radius`	(Optional) Selects the security server to use for AAA preauthentication requests. • The default is RADIUS.
Step 5	**dnis** [**password** *string*] **Example:** `Device(config-preauth)# dnis password dnispass`	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.
Step 6	**end** **Example:** `Device(config-preauth)# end`	Exits AAA preauthentication configuration mode and returns to privileged EXEC mode.

Configuring AAA Server Group Selection Based on DNIS

To configure the device to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with a DNIS number, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa dnis map enable**
4. **aaa dnis map** *dnis-number* **authentication ppp group** *server-group-name*
5. **aaa dnis map** *dnis-number* **authorization network group** *server-group-name*
6. **aaa dnis map** *dnis-number* **accounting network** [**none** | **start-stop** | **stop-only**] **group** *server-group-name*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa dnis map enable** **Example:** `Device(config)# aaa dnis map enable`	Enables DNIS mapping.
Step 4	**aaa dnis map** *dnis-number* **authentication ppp group** *server-group-name* **Example:** `Device(config)# aaa dnis map 7777 authentication ppp group sg1`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
Step 5	**aaa dnis map** *dnis-number* **authorization network group** *server-group-name* **Example:** `Device(config)# aaa dnis map 7777 authorization network group sg1`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authorization.
Step 6	**aaa dnis map** *dnis-number* **accounting network** [**none** | **start-stop** | **stop-only**] **group** *server-group-name* **Example:** `Device(config)# aaa dnis map 8888 accounting network stop-only group sg2`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring AAA Preauthentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa preauthorization**

4. **group** *server-group*
5. **clid** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
6. **ctype** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
7. **dnis** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
8. **dnis bypass** *dnis-group-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa preauthorization** **Example:** `Device(config)# aaa preauthorization`	Enters AAA preauthentication configuration mode.
Step 4	**group** *server-group* **Example:** `Device(config-preauth)# group sg2`	Specifies the AAA RADIUS server group to use for preauthentication.
Step 5	**clid** [**if-avail**	**required**] [**accept-stop**] [**password** *string*] **Example:** `Device(config-preauth)# clid required`	Preauthenticates calls on the basis of the CLID number.
Step 6	**ctype** [**if-avail**	**required**] [**accept-stop**] [**password** *string*] **Example:** `Device(config-preauth)# ctype required`	Preauthenticates calls on the basis of the call type.
Step 7	**dnis** [**if-avail**	**required**] [**accept-stop**] [**password** *string*] **Example:** `Device(config-preauth)# dnis required`	Preauthenticates calls on the basis of the DNIS number.
Step 8	**dnis bypass** *dnis-group-name* **Example:**	Specifies a group of DNIS numbers that will be bypassed for preauthentication.

	Command or Action	Purpose
	`Device(config-preauth)# dnis bypass group1`	
Step 9	**end** **Example:** `Device(config-preauth)# end`	Exits preauthentication configuration mode and returns to privileged EXEC mode.

Configuring a Guard Timer

To set a guard timer to accept or reject a call in the event that the RADIUS server fails to respond to an authentication or preauthentication request, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **isdn guard-timer** *milliseconds* [**on-expiry** {**accept** | **reject**}]
5. **call guard-timer** *milliseconds* [**on-expiry** {**accept** | **reject**}]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 1/0/0:23`	Enters interface configuration mode.
Step 4	**isdn guard-timer** *milliseconds* [**on-expiry** {**accept** | **reject**}] **Example:** `Device(config-if)# isdn guard-timer 8000 on-expiry reject`	Sets an ISDN guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.

	Command or Action	Purpose
Step 5	**call guard-timer** *milliseconds* [**on-expiry** {**accept** | **reject**}] **Example:** `Device(config-if)# call guard-timer 2000 on-expiry accept`	Sets a CAS guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for AAA DNIS Map for Authorization

Example: AAA Server Group Selection Based on DNIS

The following example shows how to select RADIUS server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the RADIUS attributes for each server
! that will be associated with one of the defined server groups.
radius-server host 172.16.0.1 auth-port 1645 acct-port 1646 key cisco1
radius-server host 172.17.0.1 auth-port 1645 acct-port 1646 key cisco2
radius-server host 172.18.0.1 auth-port 1645 acct-port 1646 key cisco3
radius-server host 172.19.0.1 auth-port 1645 acct-port 1646 key cisco4
radius-server host 172.20.0.1 auth-port 1645 acct-port 1646 key cisco5
! The following commands define the sg1 RADIUS server group and associate servers
! with it.
aaa group server radius sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 RADIUS server group and associate a server
! with it.
aaa group server radius sg2
  server 172.18.0.1
! The following commands define the sg3 RADIUS server group and associate a server
! with it.
aaa group server radius sg3
  server 172.19.0.1
! The following commands define the default-group RADIUS server group and associate
! a server with it.
aaa group server radius default-group
  server 172.20.0.1
! The next set of commands configures default-group RADIUS server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using
```

```
! DNIS 7777 are sent to the sg1 server group. The accounting records for these
! connections (specifically, start-stop records) are handled by the sg2 server group.
! Calls with a DNIS of 8888 use server group sg3 for authentication and server group
! default-group for accounting. Calls with a DNIS of 9999 use server group
! default-group for authentication and server group sg3 for accounting records
! (stop records only). All other calls with DNIS other than the ones defined use the
! server group default-group for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

Examples: AAA Preauthentication

The following is a simple configuration that specifies that the DNIS number be used for preauthentication:

```
aaa preauthentication
 group radius
 dnis required
```

The following example shows a configuration that specifies that both the DNIS number and the CLID number be used for preauthentication. DNIS preauthentication is performed first, followed by CLID preauthentication.

```
aaa preauthentication
 group radius
 dnis required
 clid required
```

The following example specifies that preauthentication be performed on all DNIS numbers except the two DNIS numbers specified in the DNIS group called “dnis-group1”:

```
aaa preauthentication
 group radius
 dnis required
 dnis bypass dnis-group1
dialer dnis group dnis-group1
 number 12345
 number 12346
```

The following is a sample AAA configuration with DNIS preauthentication:

```
aaa new-model
aaa authentication login CONSOLE none
aaa authentication login RADIUS_LIST group radius
aaa authentication login TAC_PLUS group tacacs+ enable
aaa authentication login V.120 none
aaa authentication enable default enable group tacacs+
aaa authentication ppp RADIUS_LIST if-needed group radius
aaa authorization exec RADIUS_LIST group radius if-authenticated
aaa authorization exec V.120 none
aaa authorization network default group radius if-authenticated
aaa authorization network RADIUS_LIST if-authenticated group radius
aaa authorization network V.120 group radius if-authenticated
aaa accounting suppress null-username
aaa accounting exec default start-stop group radius
aaa accounting commands 0 default start-stop group radius
aaa accounting network default start-stop group radius
aaa accounting connection default start-stop group radius
aaa accounting system default start-stop group radius
```

```
aaa preauthentication
 dnis password Cisco-DNIS
aaa nas port extended
!
radius-server configure-nas
radius-server host 10.0.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.255.255.255 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 2
radius-server deadtime 1
radius-server attribute nas-port format c
radius-server unique-ident 18
radius-server key MyKey
```

Note To configure preauthentication, you must also set up preauthentication profiles on the RADIUS server.

Examples: Guard Timer for ISDN and CAS

The following example shows an ISDN guard timer that is set at 8000 milliseconds. A call is rejected if the RADIUS server does not respond to a preauthentication request when the timer expires.

```
interface serial 1/0/0:23
 isdn guard-timer 8000 on-expiry reject
aaa preauthentication
 group radius
 dnis required
```

The following example shows a CAS guard timer that is set at 20,000 milliseconds. A call is accepted if the RADIUS server does not respond to a preauthentication request when the timer expires.

```
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb dtmf dnis
 cas-custom 0
 call guard-timer 20000 on-expiry accept
aaa preauthentication
group radius
dnis required
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Related Topic	Document Title
AAA	*Authentication, Authorization, and Accounting Configuration Guide* (part of the Securing User Services Configuration Library)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

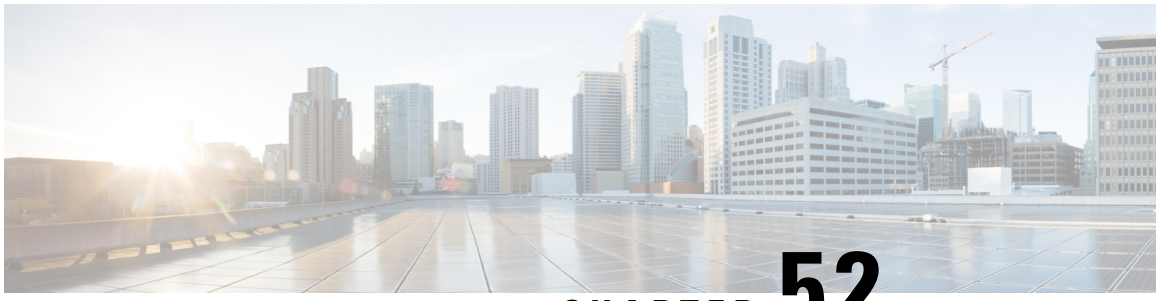
Feature Information for AAA DNIS Map for Authorization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 76: Feature Information for AAA DNIS Map for Authorization

Feature Name	Releases	Feature Information
AAA DNIS Map for Authorization	12.1(1)T 12.2(2)T 12.2(27)SBA Cisco IOS XE Release 2.3	The AAA DNIS Map for Authorization feature allows you to assign a Dialed Number Identification Service (DNIS) number to a particular AAA server group so that the server group can process authentication, authorization, and accounting requests for users dialing in to the network using that particular DNIS. Any phone line (a regular home phone or a commercial T1/ PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you. The following commands were introduced or modified: **aaa dnis enable**, **aaa dnis map authentication group**, **aaa dnis map authorization network group**, and **aaa dnis map accounting network**.



CHAPTER 52

AAA Server Groups

Configuring a device to use authentication, authorization, and accounting (AAA) server groups provides a way to group existing server hosts. Grouping existing server hosts allows you to select a subset of the configured server hosts and use them for a particular service. Configuring deadtime within a server group allows you to direct AAA traffic to separate groups of servers that have different operational characteristics. This feature module describes how to configure AAA server groups and the deadtimer.

Information About AAA Server Groups

AAA Server Groups

Configuring the device to use AAA server groups provides a way to group existing server hosts. Grouping existing server hosts allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups can also include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry that is configured acts as a failover backup to the first one. If the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order in which they are configured.)

AAA Server Groups with a Deadtimer

After you configure a server host with a server name, you can use the **deadtime** command to configure each server per server group. Configuring deadtime within a server group allows you to direct AAA traffic to separate groups of servers that have different operational characteristics.

Configuring deadtime is not limited to a global configuration. A separate timer is attached to each server host in every server group. Therefore, when a server is found to be unresponsive after numerous retransmissions and timeouts, the server is assumed to be dead. The timers attached to each server host in all server groups are triggered. In essence, the timers are checked and subsequent requests to a server (once it is assumed to be dead) are directed to alternate timers, if configured. When the network access server receives a reply from the server, it checks and stops all configured timers (if running) for that server in all server groups.

If the timer has expired, the server to which the timer is attached is assumed to be alive. This becomes the only server that can be tried for later AAA requests using the server groups to which the timer belongs.

Note Because one server has different timers and might have different deadtime values configured in the server groups, the same server might, in the future, have different states (dead and alive) at the same time.

Note To change the state of a server, you must start and stop all configured timers in all server groups.

The size of the server group will be slightly increased because of the addition of new timers and the deadtime attribute. The overall impact of the structure depends on the number and size of the server groups and how the servers are shared among server groups in a specific configuration.

How to Configure AAA Server Groups

Configuring AAA Server Groups

To define a server host with a server group name, enter the following commands in global configuration mode. The listed server must exist in global configuration mode.

Before you begin

Each server in the group must be defined previously using the **radius-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server-name*
4. **aaa group server** {**radius** | **tacacs+**} *group-name*
5. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *server-name* **Example:** `Device(config)# radius server rad1`	Specifies the name for the RADIUS server.
Step 4	**aaa group server** {**radius** | **tacacs+**} *group-name* **Example:** `Device(config)# aaa group server radius group1`	Defines the AAA server group with a group name. • All members of a group must be the same type, that is, RADIUS or TACACS+. This command puts the device in server group RADIUS configuration mode.
Step 5	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Device(config-sg-radius)# server 172.16.1.1 acct-port 1616`	Associates a particular RADIUS server with the defined server group. • Each security server is identified by its IP address and UDP port number. • Repeat this step for each RADIUS server in the AAA server group.
Step 6	**end** **Example:** `Device(config-sg-radius)# end`	Exits server group RADIUS configuration mode and returns to privileged EXEC mode.

Configuring AAA Server Groups with a Deadtimer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group*
4. **deadtime** *minutes*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group* **Example:** `Device(config)# aaa group server radius group1`	Defines a RADIUS type server group and enters server group RADIUS configuration mode.
Step 4	**deadtime** *minutes* **Example:** `Device(config-sg-radius)# deadtime 1`	Configures and defines a deadtime value in minutes. **Note** Local server group deadtime overrides the global configuration. If the deadtime vlaue is omitted from the local server group configuration, it is inherited from the primary list.
Step 5	**end** **Example:** `Device(config-sg-radius)# end`	Exits the server group RADIUS configuration mode and returns to the privileged EXEC mode.

Configuration Examples for AAA Server Groups

Examples: AAA Server Groups

The following example shows how to create server group radgroup1 with three different RADIUS server members, each using the default authentication port (1645) and accounting port (1646):

```
aaa group server radius radgroup1
 server 172.16.1.11
 server 172.17.1.21
 server 172.18.1.31
```

The following example shows how to create server group radgroup2 with three RADIUS server members, each with the same IP address but with unique authentication and accounting ports:

```
aaa group server radius radgroup2
 server 172.16.1.1 auth-port 1000 acct-port 1001
```

```
 server 172.16.1.1 auth-port 2000 acct-port 2001
 server 172.16.1.1 auth-port 3000 acct-port 3001
```

Example: Multiple RADIUS Server Entries Using AAA Server Groups

The following example shows how to configure the network access server to recognize two different RADIUS server groups. One of these groups, group1, has two different host entries on the same RADIUS server configured for the same services. The second host entry configured acts as failover backup to the first one. Each group is individually configured for the deadtime; the deadtime for group 1 is one minute, and the deadtime for group 2 is two minutes.

Note In cases where both global commands and **server** commands are used, the **server** command takes precedence over the global command.

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group group1
! The following commands define the group1 RADIUS server group and associate servers
! with it and configures a deadtime of one minute.
aaa group server radius group1
 server 10.1.1.1 auth-port 1645 acct-port 1646
 server 10.2.2.2 auth-port 2000 acct-port 2001
 deadtime 1
! The following commands define the group2 RADIUS server group and associate servers
! with it and configures a deadtime of two minutes.
aaa group server radius group2
 server 10.2.2.2 auth-port 2000 acct-port 2001
 server 10.3.3.3 auth-port 1645 acct-port 1646
 deadtime 2
! The following set of commands configures the RADIUS attributes for each host entry
! associated with one of the defined server groups.
radius-server host 10.1.1.1 auth-port 1645 acct-port 1646
radius-server host 10.2.2.2 auth-port 2000 acct-port 2001
radius-server host 10.3.3.3 auth-port 1645 acct-port 1646
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
AAA and RADIUS commands	*Cisco IOS Security Command Reference*
RADIUS attributes	*RADIUS Attributes Configuration Guide* (part of the Securing User Services Configuration Library)
AAA	*Authentication, Authorization, and Accounting Configuration Guide* (part of the Securing User Services Configuration Library)

Related Topic	Document Title
L2TP, VPN, or VPDN	*Dial Technologies Configuration Guide* and *VPDN Configuration Guide*
Modem configuration and management	*Dial Technologies Configuration Guide*
RADIUS port identification for PPP	*Wide-Area Networking Configuration Guide*

RFCs

RFC	Title
RFC 2138	*Remote Authentication Dial-In User Service (RADIUS)*
RFC 2139	*RADIUS Accounting*
RFC 2865	*RADIUS*
RFC 2867	*RADIUS Accounting Modifications for Tunnel Protocol Support*
RFC 2868	*RADIUS Attributes for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

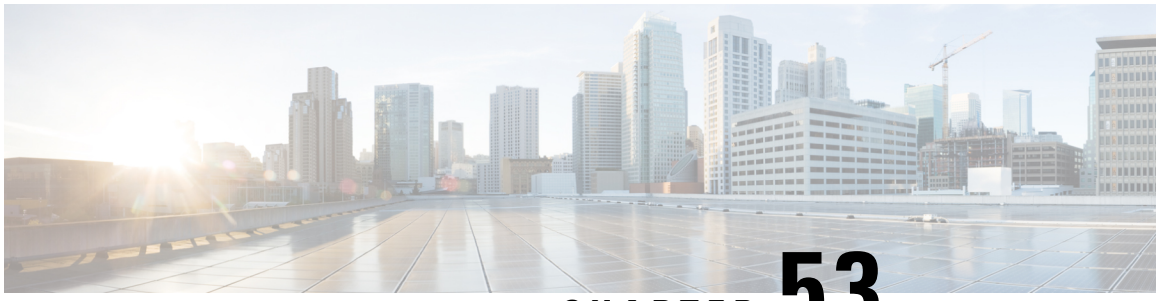
Feature Information for AAA Server Groups

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 77: Feature Information for AAA Server Groups

Feature Name	Releases	Feature Information
AAA Server Group		Configuring the device to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used with a global server-host list. The server group lists the IP addresses of the selected server hosts. • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller • Catalyst 3650 Series Switches The following commands were introduced or modified: **aaa group server radius**, **aaa group server tacacs+**, and **server** (RADIUS).
AAA Server Group Enhancements		AAA Server Group Enhancements enables the full configuration of a server in a server group. • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller • Catalyst 3650 Series Switches
AAA Server Group Deadtimer		Configuring deadtime within a server group allows you to direct AAA traffic to separate groups of servers that have different operational characteristics. • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller • Catalyst 3650 Series Switches The following commands were introduced or modified: **deadtime**.



CHAPTER 53

Framed-Route in RADIUS Accounting

The Framed-Route in RADIUS Accounting feature provides for the presence of Framed-Route (RADIUS attribute 22) information in RADIUS Accounting-Request accounting records. The Framed-Route information is returned to the RADIUS server in the Accounting-Request packets. The Framed-Route information can be used to verify that a per-user route or routes have been applied for a particular static IP customer on the network access server (NAS).

Prerequisites for Framed-Route in RADIUS Accounting

Be familiar with configuring authentication, authorization, and accounting (AAA), RADIUS servers, and RADIUS attribute screening.

Information About Framed-Route in RADIUS Accounting

Framed-Route Attribute 22

Framed-Route, attribute 22 as defined in Internet Engineering Task Force (IETF) standard RFC 2865, provides for routing information to be configured for the user on the NAS. The Framed-Route attribute information is usually sent from the RADIUS server to the NAS in Access-Accept packets. The attribute can appear multiple times.

Framed-Route in RADIUS Accounting Packets

The Framed-Route attribute information in RADIUS accounting packets shows per-user routes that have been applied for a particular static IP customer on the NAS. The Framed-Route attribute information is currently sent in Access-Accept packets. The Framed-Route attribute information is also sent in Accounting-Request

packets if it was provided in the Access-Accept packets and was applied successfully. Zero or more instances of the Framed-Route attribute may be present in the Accounting-Request packets.

Note If there is more than one Framed-Route attribute in an Access-Accept packet, there can also be more than one Framed-Route attribute in the Accounting-Request packet.

The Framed-Route information is returned in Stop and Interim accounting records and in Start accounting records when accounting Delay-Start is configured.

No configuration is required to have the Frame-Route attribute information returned in the RADIUS accounting packets.

How to Monitor Framed-Route in RADIUS Accounting

Use the **debug radius** command to monitor whether Framed-Route (attribute 22) information is being sent in RADIUS Accounting-Request packets.

Configuration Examples for Framed-Route in RADIUS Accounting

debug radius Command Output Example

In the following example, the **debug radius** command is used to verify that Framed-Route (attribute 22) information is being sent in the Accounting-Request packets (see the line 00:06:23: RADIUS: Framed-Route [22] 26 "10.80.0.1 255.255.255.255 10.60.0.1 100").

```
Router# debug radius
00:06:23: RADIUS:  Send to unknown id 0 10.1.0.2:1645, Access-Request, len 126
00:06:23: RADIUS:  authenticator 40 28 A8 BC 76 D4 AA 88 - 5A E9 C5 55 0E 50 84 37
00:06:23: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
00:06:23: RADIUS:  User-Name           [1]   14  "nari@trw1001"
00:06:23: RADIUS:  CHAP-Password       [3]   19  *
00:06:23: RADIUS:  NAS-Port            [5]   6   1
00:06:23: RADIUS:  Vendor, Cisco       [26]  33
00:06:23: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:23: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
00:06:23: RADIUS:  Service-Type        [6]   6   Framed                    [2]
00:06:23: RADIUS:  NAS-IP-Address      [4]   6   12.1.0.1
00:06:23: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
00:06:23: RADIUS:  Received from id 0 10.1.0.2:1645, Access-Accept, len 103
00:06:23: RADIUS:  authenticator 5D 2D 9F 25 11 15 45 B2 - 54 BB 7F EB CE 79 20 3B
00:06:23: RADIUS:  Vendor, Cisco       [26]  33
00:06:23: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:23: RADIUS:  Service-Type        [6]   6   Framed                    [2]
00:06:23: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
00:06:23: RADIUS:  Framed-IP-Netmask   [9]   6   255.255.255.255
00:06:23: RADIUS:  Framed-IP-Address   [8]   6   10.60.0.1
00:06:23: RADIUS:  Framed-Route        [22]  26  "10.80.0.1 255.255.255.255 10.60.0.1 100"
        <=======
00:06:23: RADIUS:  Received from id 2
00:06:24: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state
to up
```

```
00:06:25: AAA/AUTHOR:  Processing PerUser AV route
00:06:25: Vi1 AAA/PERUSER/ROUTE: route string: IP route  10.80.0.1 255.255.255.255 10.60.0.1
 100
00:06:25: RADIUS/ENCODE(00000002): Unsupported AAA attribute timezone
00:06:25: RADIUS(00000002): sending
00:06:25: RADIUS:  Send to unknown id 1 10.1.0.2:1646, Accounting-Request, len 278
00:06:25: RADIUS:  authenticator E0 CC 99 EB 49 18 B9 78 - 4A 09 60 0F 4E 92 24 C6
00:06:25: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
00:06:25: RADIUS:  Tunnel-Server-Endpoi[67]  12  00:"10.1.1.1"
00:06:25: RADIUS:  Tunnel-Client-Endpoi[66]  12  00:"10.1.1.2"
00:06:25: RADIUS:  Tunnel-Assignment-Id[82]  15  00:"from_isdn101"
00:06:25: RADIUS:  Tunnel-Type         [64]  6   00:L2TP                    [3]
00:06:25: RADIUS:  Acct-Tunnel-Connecti[68]  12  "2056100083"
00:06:25: RADIUS:  Tunnel-Client-Auth-I[90]  10  00:"isdn101"
00:06:25: RADIUS:  Tunnel-Server-Auth-I[91]  6   00:"lns"
00:06:25: RADIUS:  Framed-Protocol     [7]   6   PPP                        [1]
00:06:25: RADIUS:  Framed-Route        [22]  39  "10.80.0.1 255.255.255.255 10.60.0.1 100"
        <========
00:06:25: RADIUS:  Framed-IP-Address   [8]   6   10.60.0.1
00:06:25: RADIUS:  Vendor, Cisco       [26]  35
00:06:25: RADIUS:  Cisco AVpair        [1]   29  "connect-progress=LAN Ses Up"
00:06:25: RADIUS:  Authentic           [45]  6   RADIUS                     [1]
00:06:25: RADIUS:  User-Name           [1]   14  "username1@example.com"
00:06:25: RADIUS:  Acct-Status-Type    [40]  6   Start                      [1]
00:06:25: RADIUS:  NAS-Port            [5]   6   1
00:06:25: RADIUS:  Vendor, Cisco       [26]  33
00:06:25: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:25: RADIUS:  NAS-Port-Type       [61]  6   Virtual                    [5]
00:06:25: RADIUS:  Service-Type        [6]   6   Framed                     [2]
00:06:25: RADIUS:  NAS-IP-Address      [4]   6   10.1.0.1
00:06:25: RADIUS:  Acct-Delay-Time     [41]  6   0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
RADIUS	“Configuring RADIUS” feature module.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	Remote Authentication Dial In User Service (RADIUS)
RFC 3575	IANA Considerations for RADIUS (Remote Authentication Dial In User Service)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

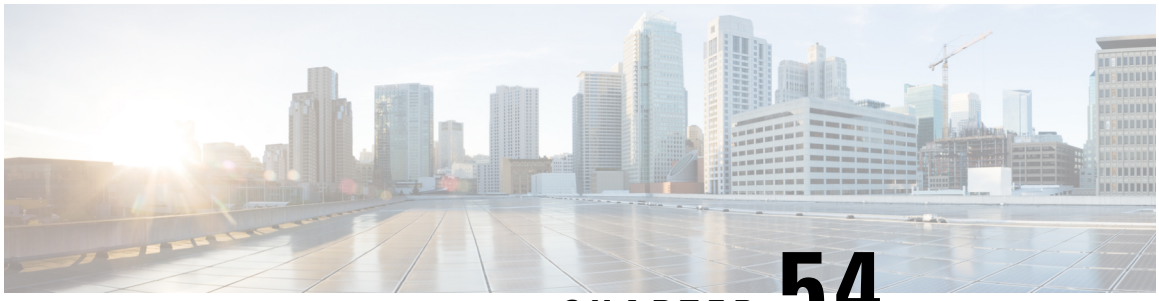
Feature Information for Framed-Route in RADIUS Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 78: Feature Information for Framed-Route in RADIUS Accounting

Feature Name	Releases	Feature Information
Framed-Route in RADIUS Accounting	Cisco IOS XE Release 2.1	The Framed-Route in RADIUS Accounting feature provides for the presence of Framed-Route (RADIUS attribute 22) information in RADIUS Accounting-Request accounting records. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 54

RFC-2867 RADIUS Tunnel Accounting

The RFC-2867 RADIUS Tunnel Accounting introduces six new RADIUS accounting types that are used with the RADIUS accounting attribute Acct-Status-Type (attribute 40), which indicates whether an accounting request marks the beginning of user service (start) or the end (stop).

This feature also introduces two new virtual private virtual private dialup network (VPDN) commands that help users better troubleshoot VPDN session events.

Restrictions for RFC-2867 RADIUS Tunnel Accounting

RADIUS tunnel accounting works only with L2TP tunnel support.

Information About RFC-2867 RADIUS Tunnel Accounting

Benefits of RFC-2867 RADIUS Tunnel Accounting

Without RADIUS tunnel accounting support, VPDN with network accounting, which allows users to determine tunnel-link status changes, did not report all possible attributes to the accounting record file. Now that all possible attributes can be displayed, users can better verify accounting records with their Internet Service Providers (ISPs).

RADIUS Attributes Support for RADIUS Tunnel Accounting

The table below outlines the new RADIUS accounting types that are designed to support the provision of compulsory tunneling in dialup networks; that is, these attribute types allow you to better track tunnel status changes.

Note The accounting types are divided into two separate tunnel types so users can decide if they want tunnel type, tunnel-link type, or both types of accounting.

Table 79: RADIUS Accounting Types for the Acct-Status-Type Attribute

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Start	9	Marks the beginning of a tunnel setup with another node.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • Acct-Delay-Time (41)--from AAA • Event-Timestamp (55)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Stop	10	Marks the end of a tunnel connection to or from another node.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • Acct-Delay-Time (41)--from AAA • Acct-Input-Octets (42)--from AAA • Acct-Output-Octets (43)--from AAA • Acct-Session-Id (44)--from AAA • Acct-Session-Time (46)--from AAA • Acct-Input-Packets (47)--from AAA • Acct-Output-Packets (48)--from AAA • Acct-Terminate-Cause (49)--from AAA • Acct-Multi-Session-Id (51)--from AAA • Event-Timestamp (55)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client • Acct-Tunnel-Packets-Lost (86)--from client

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Reject	11	Marks the rejection of a tunnel setup with another node.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • Acct-Delay-Time (41)--from AAA • Acct-Terminate-Cause (49)--from client • Event-Timestamp (55)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client
Tunnel-Link-Start	12	Marks the creation of a tunnel link. Only some tunnel types (Layer 2 Transport Protocol [L2TP]) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • NAS-Port (5)--from AAA • Acct-Delay-Time (41)--from AAA • Event-Timestamp (55)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Link-Stop	13	Marks the end of a tunnel link. Only some tunnel types (L2TP) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • NAS-Port (5)--from AAA • Acct-Delay-Time (41)--from AAA • Acct-Input-Octets (42)--from AAA • Acct-Output-Octets (43)--from AAA • Acct-Session-Id (44)--from AAA • Acct-Session-Time (46)--from AAA • Acct-Input-Packets (47)--from AAA • Acct-Output-Packets (48)--from AAA • Acct-Terminate-Cause (49)--from AAA • Acct-Multi-Session-Id (51)--from AAA • Event-Timestamp (55)--from AAA • NAS-Port-Type (61)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client • Acct-Tunnel-Packets-Lost (86)--from client

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Link-Reject	14	Marks the rejection of a tunnel setup for a new link in an existing tunnel. Only some tunnel types (L2TP) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)--from client • NAS-IP-Address (4)--from AAA • Acct-Delay-Time (41)--from AAA • Acct-Terminate-Cause (49)--from AAA • Event-Timestamp (55)--from AAA • Tunnel-Type (64)--from client • Tunnel-Medium-Type (65)--from client • Tunnel-Client-Endpoint (66)--from client • Tunnel-Server-Endpoint (67)--from client • Acct-Tunnel-Connection (68)--from client

[1] If the specified tunnel type is used, these attributes should also be included in the accounting request packet.

How to Configure RADIUS Tunnel Accounting

Enabling Tunnel Type Accounting Records

Use this task to configure your LAC to send tunnel and tunnel-link accounting records to be sent to the RADIUS server.

Two new command line interfaces (CLIs)--vpdn session accounting network(tunnel-link-type records)and vpdn tunnel accounting network(tunnel-type records) --are supported to help identify the following events:

- A VPDN tunnel is brought up or destroyed
- A request to create a VPDN tunnel is rejected
- A user session within a VPDN tunnel is brought up or brought down
- A user session create request is rejected

Note The first two events are tunnel-type accounting records: authentication, authorization, and accounting (AAA) sends Tunnel-Start, Tunnel-Stop, or Tunnel-Reject accounting records to the RADIUS server. The next two events are tunnel-link-type accounting records: AAA sends Tunnel-Link-Start, Tunnel-Link-Stop, or Tunnel-Link-Reject accounting records to the RADIUS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **aaa accounting network default** *list-name*} {**start-stop** | **stop-only** | **wait-start** | **none group** *groupname*
4. Router(config)# **vpdn enable**
5. Router(config)# **vpdn tunnel accounting network** *list-name*
6. Router(config)# **vpdn session accounting network** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **aaa accounting network default** *list-name*} {**start-stop** | **stop-only** | **wait-start** | **none group** *groupname* **Example:** **Example:** **Example:** **Example:** **Example:** **Example:** **Example:** **Example:**	Enables network accounting. • **default** --If the default network accounting method-list is configured and no additional accounting configurations are enabled on the interface, network accounting is enabled by default. If either the **vpdn session accounting network** command or the **vpdn tunnel accounting network** command is linked to the **default** method-list, all tunnel and tunnel-link accounting records are enabled for those sessions. • *list-name* --The *list-name*defined in the **aaa accounting**command must be the same as the *list-name* defined in the VPDN command; otherwise, accounting will not occur.

	Command or Action	Purpose
	Example: **Example:** `Router(config)# aaa accounting network m1 start-stop group radius`	
Step 4	Router(config)# **vpdn enable** **Example:** `Router(config)# vpdn enable`	Enables virtual private dialup networking on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server (if applicable).
Step 5	Router(config)# **vpdn tunnel accounting network** *list-name* **Example:** `Router(config)# vpdn tunnel accounting network m1`	Enables Tunnel-Start, Tunnel-Stop, and Tunnel-Reject accounting records. • *list-name* --The *list-name* must match the *list-name* defined in the **aaa accounting**command; otherwise, network accounting will not occur.
Step 6	Router(config)# **vpdn session accounting network** *list-name* **Example:** `Router(config)# vpdn session accounting network m1`	Enables Tunnel-Link-Start, Tunnel-Link-Stop, and Tunnel-Link-Reject accounting records. • *list-name* --The *list-name* must match the *list-name* defined in the **aaa accounting**command; otherwise, network accounting will not occur.

What To Do Next

After you have enabled RADIUS tunnel accounting, you can verify your configuration via the following optional task Verifying RADIUS Tunnel Accounting.

Verifying RADIUS Tunnel Accounting

Use either one or both of the following optional steps to verify your RADIUS tunnel accounting configuration.

SUMMARY STEPS

1. **enable**
2. Router# **show accounting**
3. Router# **show vpdn** [**session**] [**tunnel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	Router# **show accounting** **Example:** `Router# show accounting`	Displays the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.
Step 3	Router# **show vpdn** [**session**] [**tunnel**] **Example:** **Example:** **Example:** **Example:** `Router# show vpdn session`	Displays information about active L2TP tunnel and message identifiers in a VPDN. • **session** --Displays a summary of the status of all active tunnels. • **tunnel** --Displays information about all active L2TP tunnels in summary-style format.

Configuration Examples for RADIUS Tunnel Accounting

Configuring RADIUS Tunnel Accounting on LAC Example

The following example shows how to configure your L2TP access concentrator (LAC) to send tunnel and tunnel-link accounting records to the RADIUS server:

```
aaa new-model
!
!
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
aaa session-id common
enable secret 5 $1$IDjH$iL7puCja1RMlyOM.JAeuf/
enable password lab
!
username ISP_LAC password 0 tunnelpass
!
!
resource-pool disable
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip host dirt 172.16.1.129
!
vpdn enable
```

```
vpdn tunnel accounting network m1
vpdn session accounting network m1
vpdn search-order domain dnis
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain cisco.com
 initiate-to ip 10.1.26.71
 local name ISP_LAC
!
mta receive maximum-recipients 0
!
interface GigabitEthernet0/0/0
 ip address 10.1.27.74 255.255.255.0
 no ip mroute-cache
 duplex half
 speed auto
 no cdp enable
!
interface FastEthernet0/0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip default-gateway 10.1.27.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.27.254
no ip http server
ip pim bidir-enable
!
no cdp run
!
!
radius-server host 172.19.192.26 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
!
```

Configuring RADIUS Tunnel Accounting on LNS Example

The following example shows how to configure your L2TP network server (LNS) to send tunnel and tunnel-link accounting records to the RADIUS server:

```
aaa new-model
!
!
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
aaa session-id common
enable secret 5 $1$ftf.$wE6Q5Yv6hmQiwL9pizPCg1
!
username ENT_LNS password 0 tunnelpass
username user1@cisco.com password 0 lab
username user2@cisco.com password 0 lab
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/9
 firmware location system:/ucode/mica_port_firmware
```

```
!
!
resource-pool disable
clock timezone est 2
!
ip subnet-zero
no ip domain-lookup
ip host CALLGEN-SECURITY-V2 172.24.80.28 10.47.0.0
ip host dirt 172.16.1.129
!
vpdn enable
vpdn tunnel accounting network m1
vpdn session accounting network m1
!
vpdn-group 1
accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname ISP_LAC
 local name ENT_LNS
!
mta receive maximum-recipients 0
!
interface Loopback0
 ip address 192.168.70.101 255.255.255.0
!
interface Loopback1
 ip address 192.168.80.101 255.255.255.0
!
interface FastEthernet0/0/0
 ip address 10.1.26.71 255.255.255.0
 no ip mroute-cache
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Loopback0
 peer default ip address pool vpdn-pool1
 ppp authentication chap
!
interface Virtual-Template2
 ip unnumbered Loopback1
 peer default ip address pool vpdn-pool2
 ppp authentication chap
!
interface FastEthernet0/0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip local pool vpdn-pool1 192.168.70.1 192.168.70.100
ip local pool vpdn-pool2 192.168.80.1 192.168.80.100
ip default-gateway 10.1.26.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.26.254
ip route 10.90.1.2 255.255.255.255 10.1.26.254
no ip http server
ip pim bidir-enable
!
no cdp run
!
radius-server host 172.19.192.80 auth-port 1645 acct-port 1646 key rad123
```

```
radius-server retransmit 3
call rsvp-sync
```

Additional References

The following sections provide references related to RFC-2867 RADIUS Tunnel Accounting.

Related Documents

Related Topic	Document Title
RADIUS attributes	"RADIUS Attributes Overview and RADIUS IETF Attributes" in the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
VPDN	*Cisco IOS XE VPDN Configuration Guide* , Release 2
Network accounting	"Configuring Accounting" in the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
Commands	• *Cisco IOS Security Command Reference* • *Cisco IOS VPDN Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2867	*RADIUS Accounting Modifications for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

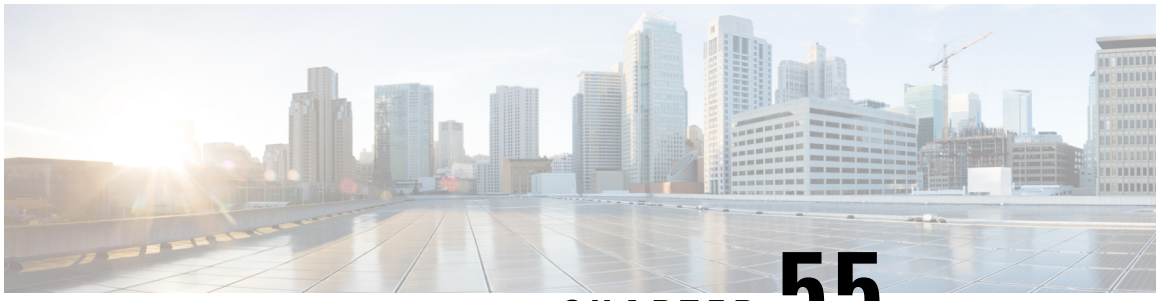
Feature Information for RFC-2867 RADIUS Tunnel Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 80: Feature Information for RFC-2867 RADIUS Tunnel Accounting

Feature Name	Releases	Feature Information
RFC-2867 RADIUS Tunnel Accounting	Cisco IOS XE Release 2.1	The RFC-2867 RADIUS Tunnel Accounting introduces six new RADIUS accounting types that are used with the RADIUS accounting attribute Acct-Status-Type (attribute 40), which indicates whether an accounting request marks the beginning of user service (start) or the end (stop). This feature also introduces two new virtual private virtual private dialup network (VPDN) commands that help users better troubleshoot VPDN session events. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **aaa accounting**, **vpdn session accounting network**, **vpdn tunnel accounting network**.



CHAPTER 55

RADIUS Logical Line ID

The RADIUS Logical Line ID feature, also known as the Logical Line Identification (LLID) Blocking feature enables administrators to track their customers on the basis of the physical lines on which customer calls originate. Administrators use a virtual port that does not change as customers move from one physical line to another. This virtual port facilitates the maintenance of the administrator's customer profile database and allows the administrator to do additional security checks on customers.

Prerequisites for RADIUS Logical Line ID

Although this feature can be used with any RADIUS server, some RADIUS servers may require modifications to their dictionary files to allow the Calling-Station-ID attribute to be returned in Access-Accept messages. For example, the Merit RADIUS server does not support LLID downloading unless you modify its dictionary as follows: "ATTRIBUTE Calling-Station-Id 31 string (*, *)"

Restrictions for RADIUS Logical Line ID

The RADIUS Logical Line ID feature supports RADIUS only. TACACS+ is not supported.

This feature can be applied only toward PPP over Ethernet over ATM (PPPoEoATM) and PPP over Ethernet over VLAN (PPPoEoVLAN) (Dot1Q) calls; no other calls, such as ISDN, can be used.

Information About RADIUS Logical Line ID

Preauthorization

LLID is an alphanumeric string (which must be a minimum of one character and a maximum of 253 characters) that is a logical identification of a subscriber line. LLID is maintained in a customer profile database on a RADIUS server. When the customer profile database receives a preauthorization request from the access router, the RADIUS server sends the LLID to the router as the Calling-Station-ID attribute (attribute 31).

The Layer 2 Tunneling Protocol (L2TP) access concentrator (LAC) sends a preauthorization request to the customer profile database when the LAC is configured for preauthorization. Configure the LAC for preauthorization using the **subscriber access** command.

Note Downloading the LLID is referred to as "preauthorization" because it occurs before either service (domain) authorization or user authentication and authorization occur.

The customer profile database on the RADIUS server consists of user profiles for each physical network access server (NAS) port that is connected to the router. Each user profile contains a profile matched to a username (attribute 1) representing the physical port on the router. When the router is configured for preauthorization, it queries the customer profile database using a username representative of the physical NAS port making the connection to the router. When a match is found in the customer profile database, the customer profile database returns an Access-Accept message containing the LLID in the user profile. The LLID is defined in the Access-Accept record as the Calling-Station-ID attribute.

The preauthorization process can also provide the real username being used for authentication to the RADIUS server. Because the physical NAS port information is being used as the username (attribute 1), RADIUS attribute 77 (Connect-Info) can be configured to contain the authentication username. This configuration allows the RADIUS server to provide additional validation on the authorization request if it chooses, such as analyzing the username for privacy rules, before returning an LLID back to the router.

How to Configure RADIUS Logical Line ID

Configuring Preauthorization

To download the LLID and configure the LAC for preauthorization, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *interface-name*
4. **subscriber access** {**pppoe** | **pppoa**} **pre-authorize nas-port-id** [**default** | *list-name*] [**send username**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *interface-name* **Example:** **Example:** `Router (config)# ip radius source-interface Loopback1`	Specifies the IP address portion of the username for the preauthorization request.
Step 4	**subscriber access** {**pppoe** | **pppoa**} **pre-authorize nas-port-id** [**default** | *list-name*] [**send username**] **Example:** **Example:** `Router (config)# subscriber access pppoe pre-authorize nas-port-id mlist_llid send username`	Enables the LLID to be downloaded so the router can be configured for preauthorization. The **send username** option specifies that you include the authentication username of the session inside the Connect-Info (attribute 77) in the Access-Request message.

Configuring the LLID in a RADIUS User Profile

To configure the user profile for preauthorization, add a NAS port user to the customer profile database and add RADIUS Internet Engineering Task Force (IETF) attribute 31 (Calling-Station-ID) to the user profile.

SUMMARY STEPS

1. UserName=nas_port: ip-address:slot/module/port/vpi.vci
2. User-Name=nas-port: ip-address:slot/module/port/vlan-id
3. Calling-Station-Id = “string (*,*)”

DETAILED STEPS

	Command or Action	Purpose
Step 1	UserName=nas_port: ip-address:slot/module/port/vpi.vci	(Optional) Adds a PPPoE over ATM NAS port user.
Step 2	User-Name=nas-port: ip-address:slot/module/port/vlan-id	(Optional) Adds a PPPoE over VLAN NAS port user.
Step 3	Calling-Station-Id = “string (*,*)”	Adds attribute 31 to the user profile. • String--One or more octets, containing the phone number from which the user placed the call.

Verifying Logical Line ID

To verify feature functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Checks to see that RADIUS attribute 31 is the LLID in the Accounting-Request on LAC and in the Access-Request and Accounting-Request on the LNS.

Configuration Examples for RADIUS Logical Line ID

LAC for Preauthorization Configuration Example

The following example shows how to configure your LAC for preauthorization by downloading the LLID:

```
aaa new-model
aaa group server radius sg_llid
 server 172.31.164.106 auth-port 1645 acct-port 1646
aaa group server radius sg_water
 server 172.31.164.106 auth-port 1645 acct-port 1646
aaa authentication ppp default group radius
aaa authorization confg-commands
```

```
aaa authorization network default group sg_water
aaa authorization network mlist_llid group sg_llid
aaa session-id common
!
username s7200_2 password 0 lab
username s5300 password 0 lab
username sg_water password 0 lab
vpdn enable
!
vpdn-group 2
 request-dialin
 protocol l2tp
 domain example.com
 domain example.com#184
 initiate-to ip 10.1.1.1
 local name s7200_2
 l2tp attribute clid mask-method right * 255 match #184
!
vpdn-group 3
 accept dialin
  protocol pppoe
  virtual-template 1
!
!
Enable the LLID to be downloaded.
subscriber access pppoe pre-authorize nas-port-id mlist_llid send username
!
interface Loopback0
 ip address 10.1.1.2 255.255.255.0
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.0
!
interface FastEthernet1/0/0
 ip address 10.1.1.8 255.255.255.0 secondary
 ip address 10.0.58.111 255.255.255.0
 no cdp enable
!
interface ATM4/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM4/0/0.1 point-to-point
 pvc 1/100
  encapsulation aa15snap
  protocol pppoe
!
interface virtual-template1
 no ip unnumbered Loopback0
 no peer default ip address
 ppp authentication chap
!
radius-server host 172.31.164.120 auth-port 1645 acct-port 1646 key rad123
radius-server host 172.31.164.106 auth-port 1645 acct-port 1646 key rad123
ip radius source-interface Loopback1
```

RADIUS User Profile for LLID Example

The following example shows how to configure the user profile for LLID querying for PPPoEoVLAN and PPPoEoATM and how to add attribute 31:

```
pppoeovlan
```

```
----------
nas-port:10.1.0.3:6/0/0/0    Password = "password1",
    Service-Type = Outbound,
    Calling-Station-ID = "cat-example"
pppoeoa
--------
nas-port:10.1.0.3:6/0/0/1.100    Password = "password1",
    Service-Type = Outbound,
    Calling-Station-ID = "cat-example"
```

Additional References

The following sections provide references related to RADIUS EAP Support feature.

Related Documents

Related Topic	Document Title
Configuring PPP Authentication Using AAA	“ Configuring Authentication ” module.
Configuring RADIUS	“ Configuring RADIUS ” module.
PPP Configuration	“ Configuring Asynchronous SLIP and PPP ” module.
Dial Technologies commands	*Cisco IOS Dial Technologies Command Reference*
Security Commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 1938	*A One-Time Password System*
RFC 2869	*RADIUS Extensions*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Logical Line ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 81: Feature Information for RADIUS Logical Line ID

Feature Name	Releases	Feature Information
RADIUS Logical Line ID	Cisco IOS XE Release 2.1	The RADIUS Logical Line ID feature, also known as the Logical Line Identification (LLID) Blocking feature enables administrators to track their customers on the basis of the physical lines on which customer calls originate. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced or modified by this feature: **subscriber access**.
Calling Station ID Attribute 31	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
LLID Blocking	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

attribute --A RADIUS Internet Engineering Task Force (IETF) attribute is one of the original set of 255 standard attributes that are used to communicate authentication, authorization, and accounting (AAA) information between a client and a server. Because IETF attributes are standard, the attribute data is predefined

and well known; thus all clients and servers that exchange AAA information through IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

CHAP --Challenge Handshake Authentication Protocol. Security feature that is supported on lines using PPP encapsulation and prevents unauthorized access. CHAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines whether that user is allowed access.

EAP --Extensible Authentication Protocol. A PPP authentication protocol that supports multiple authentication mechanisms that are negotiated during the authentication phase (instead of the Link Control Protocol [LCP] phase). EAP allows a third-party authentication server to interact with the PPP implementation through a generic interface.

LCP --link control protocol. Protocol that establishes, configures, and tests data-link connections for use by PPP.

MD5 (HMAC variant) --Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a key hashing for message authentication.

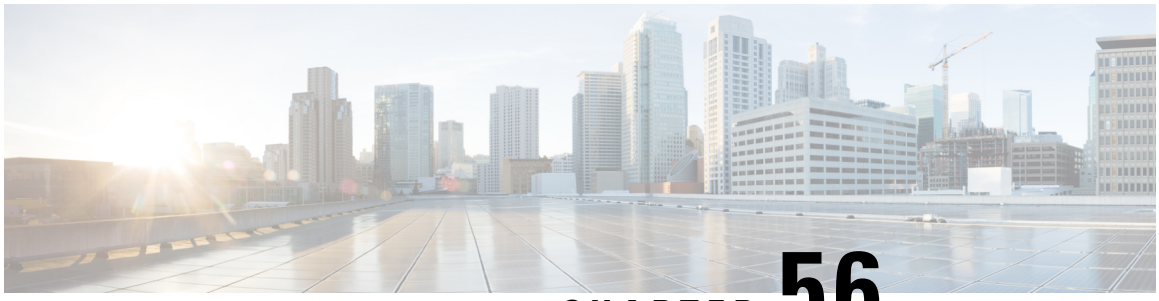
NAS --network access server. A device providing local network access to users across a remote access network such as the public switched telephone network (PSTN).

PAP --Password Authentication Protocol. Authentication protocol that allows PPP peers to authenticate one another. The remote router attempting to connect to the local router is required to send an authentication request. Unlike CHAP, PAP passes the password and host name or username in the clear (unencrypted). PAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines if that user is allowed access. PAP is supported only on PPP lines.

PPP --Point-to-Point Protocol. A protocol that encapsulates network layer protocol information over point-to-point links. PPP is defined in RFC 1661.

RADIUS --Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.



CHAPTER 56

RADIUS Route Download

The RADIUS Route Download feature allows users to configure their network access server (NAS) to direct RADIUS authorization.

Prerequisites for RADIUS Route Download

AAA network security must be enabled before you perform the tasks in this feature.

Information About RADIUS Route Download

The RADIUS Route Download feature allows users to configure their network access server (NAS) to direct RADIUS authorization. Users configure a separate named method list (in addition to the default method list) for static route download requests sent by their NAS to authorization, authentication, and accounting (AAA) servers.

Before this feature, RADIUS authorization for static route download requests was sent only to AAA servers specified by the default method list.

This feature extends the functionality of the **aaa route download** command to allow users to specify the name of the method list that will be used to direct static route download requests to the AAA servers. The **aaa route download**commandmay be used to specify a separate method list for downloading static routes. This method list can be added by using the **aaa authorization configuration** command.

How to Configure RADIUS Route Download

Configuring RADIUS Route Download

To configure the NAS to send static route download requests to the servers specified by a named method list, use the following commands in global configuration mode:

SUMMARY STEPS

1. Router(config)# **aaa authorization configuration** *method-name* **[radius | tacacs+ | group** *group-name* **]**
2. Router(config)# **aaa route download** [*time*] [**authorization** *method-list*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **aaa authorization configuration** *method-name* **[radius	tacacs+	group** *group-name* **]**	Downloads static route configuration information from the AAA server using RADIUS.
Step 2	Router(config)# **aaa route download** [*time*] [**authorization** *method-list*]	Enables the static route download feature. Use the **authorization** *method-list* attributes to specify a named method list to which RADIUS authorization requests for static route downloads are sent.

Verifying RADIUS Route Download

To verify the routes that are installed, use the **show ip route** command in EXEC mode.

To display information that is associated with RADIUS, use the **debug radius** command in privileged EXEC mode.

Configuration Examples for RADIUS Route Download

RADIUS Route Download Configuration Example

The following example shows how to configure the NAS to send static route download requests to the servers specified by the method list named “list1”:

```
aaa new-model
aaa group server radius rad1
server 10.2.2.2 auth-port 1645 acct-port 1646
!
aaa group server tacacs+ tac1
server 172.17.3.3
!
aaa authorization configuration default group radius
aaa authorization configuration list1 group rad1 group tac1
```

```
aaa route download 1 authorization list1
tacacs-server host 172.17.3.3
tacacs-server key cisco
tacacs-server administration
!
radius-server host 10.2.2.2 auth-port 1645 acct-port 1646
radius-server key cisco
```

Additional References

The following sections provide references related to RADIUS Route Download.

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

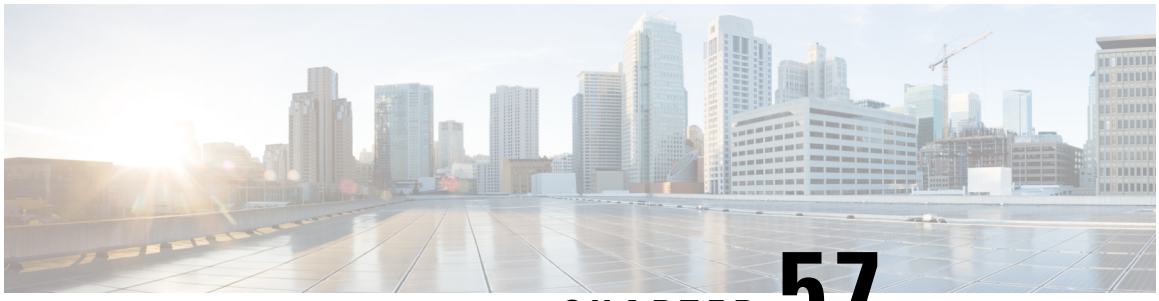
Feature Information for RADIUS Route Download

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 82: Feature Information for RADIUS Route Download

Feature Name	Releases	Feature Information
RADIUS Route Download	Cisco IOS XE Release 2.1	The RADIUS Route Download feature allows users to configure their network access server (NAS) to direct RADIUS authorization. Users configure a separate named method list (in addition to the default method list) for static route download requests sent by their NAS to authorization, authentication, and accounting (AAA) servers. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **aaa route download**



CHAPTER 57

RADIUS Server Load Balancing

The RADIUS Server Load Balancing feature distributes authentication, authorization, and accounting (AAA) authentication and accounting transactions across RADIUS servers in a server group. These servers can share the AAA transaction load and thereby respond faster to incoming requests.

This module describes the RADIUS Server Load Balancing feature.

Prerequisites for RADIUS Server Load Balancing

- Authentication, authorization, and accounting (AAA) must be configured on the RADIUS server.
- AAA RADIUS server groups must be configured.
- RADIUS must be configured for functions such as authentication, accounting, or static route download.

Restrictions for RADIUS Server Load Balancing

- Incoming RADIUS requests, such as Packet of Disconnect (POD) requests, are not supported.

Information About RADIUS Server Load Balancing

RADIUS Server Load Balancing Overview

Load balancing distributes batches of transactions to RADIUS servers within a server group. Load balancing assigns each batch of transactions to the server with the lowest number of outstanding transactions in its queue. The process of assigning a batch of transactions is as follows:

1. The first transaction is received for a new batch.
2. All server transaction queues are checked.
3. The server with the lowest number of outstanding transactions is identified.
4. The identified server is assigned the next batch of transactions.

The batch size is a user-configured parameter. Changes in the batch size may impact CPU load and network throughput. As batch size increases, CPU load decreases and network throughput increases. However, if a large batch size is used, all available server resources may not be fully utilized. As batch size decreases, CPU load increases and network throughput decreases.

Note There is no set number for large or small batch sizes. A batch with more than 50 transactions is considered large and a batch with fewer than 25 transactions is considered small.

Note If a server group contains ten or more servers, we recommend that you set a high batch size to reduce CPU load.

Transaction Load Balancing Across RADIUS Server Groups

You can configure load balancing either per-named RADIUS server group or for the global RADIUS server group. The load balancing server group must be referred to as “radius” in the authentication, authorization, and accounting (AAA) method lists. All public servers that are part of the RADIUS server group are then load balanced.

You can configure authentication and accounting to use the same RADIUS server or different servers. In some cases, the same server can be used for preauthentication, authentication, or accounting transactions for a session. The preferred server, which is an internal setting and is set as the default, informs AAA to use the same server for the start and stop record for a session regardless of the server cost. When using the preferred server setting, ensure that the server that is used for the initial transaction (for example, authentication), the preferred server, is part of any other server group that is used for a subsequent transaction (for example, accounting).

The preferred server is not used if one of the following criteria is true:

- The **load-balance method least-outstanding ignore-preferred-server** command is used.
- The preferred server is dead.

- The preferred server is in quarantine.
- The want server flag has been set, overriding the preferred server setting.

The want server flag, an internal setting, is used when the same server must be used for all stages of a multistage transaction regardless of the server cost. If the want server is not available, the transaction fails.

You can use the **load-balance method least-outstanding ignore-preferred-server** command if you have either of the following configurations:

- Dedicated authentication server and a separate dedicated accounting server
- Network where you can track all call record statistics and call record details, including start and stop records and records that are stored on separate servers

If you have a configuration where authentication servers are a superset of accounting servers, the preferred server is not used.

RADIUS Server Status and Automated Testing

The RADIUS Server Load Balancing feature considers the server status when assigning batches. Transaction batches are sent only to live servers. We recommend that you test the status of all RADIUS load-balanced servers, including low usage servers (for example, backup servers).

Transactions are not sent to a server that is marked dead. A server is marked dead until its timer expires, at which time it moves to quarantine state. A server is in quarantine until it is verified alive by the RADIUS automated tester functionality.

To determine if a server is alive and available to process transactions, the RADIUS automated tester sends a request periodically to the server for a test user ID. If the server returns an Access-Reject message, the server is alive; otherwise the server is either dead or quarantined.

A transaction sent to an unresponsive server is failed over to the next available server before the unresponsive server is marked dead. We recommend that you use the retry reorder mode for failed transactions.

When using the RADIUS automated tester, verify that the authentication, authorization, and accounting (AAA) servers are responding to the test packets that are sent by the network access server (NAS). If the servers are not configured correctly, packets may be dropped and the server erroneously marked dead.

Caution We recommend that you use a test user that is not defined on the RADIUS server for the RADIUS server automated testing to protect against security issues that may arise if the test user is not correctly configured.

Note Use the **test aaa group** command to check load-balancing transactions.

Note Starting with Cisco IOS XE Bengaluru 17.4.1 you can configure automated tester to be VRF aware. You can use the **vrf** keyword with the **automate-tester** command to enable automate-tester for a non-default VRF.

For VRF aware automate-tester to work, you must configure the **global config ipv4/ipv6 source interface** *interface-name* **vrf** *vrf-name* command.

How to Configure RADIUS Server Load Balancing

Enabling Load Balancing for a Named RADIUS Server Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group-name*
4. **server ip-address [auth-port port-number] [acct-port port-number]**
5. **load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group-name* **Example:** `Device(config)# aaa group server radius rad-sg`	Enters server group configuration mode.
Step 4	**server ip-address [auth-port port-number] [acct-port port-number]** **Example:** `Device (config-sg-radius)server 192.0.2.238 auth-port 2095 acct-port 2096`	Configures the IP address of the RADIUS server for the group server.
Step 5	**load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**] **Example:** `Device(config-sg-radius)# load-balance method least-outstanding batch-size 30`	Enables the least-outstanding load balancing for a named server group.
Step 6	**end** **Example:** `Device(config-sg)# end`	Exits server group configuration mode and enters privileged EXEC mode.

Enabling Load Balancing for a Global RADIUS Server Group

The global RADIUS server group is referred to as "radius" in the authentication, authorization, and accounting (AAA) method lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server host** {*hostname* | *ip-address*} [**test username** *name*] [**auth-port** *number*] [**ignore-auth-port**] [**acct-port** *number*] [**ignore-acct-port**] [**idle-time** *seconds*]
4. **radius-server load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**]
5. **load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server host** {*hostname* | *ip-address*} [**test username** *name*] [**auth-port** *number*] [**ignore-auth-port**] [**acct-port** *number*] [**ignore-acct-port**] [**idle-time** *seconds*] **Example:** `Device(config)# radius-server host 192.0.2.1 test username test1 idle-time 1`	Enables RADIUS automated testing.
Step 4	**radius-server load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**] **Example:** `Device(config)# radius-server load-balance method least-outstanding`	Enables the least-outstanding load balancing for the global RADIUS server group and enters server group configuration mode. • The default batch size is 25. The batch size range is from 1 to 2147483647.
Step 5	**load-balance method least-outstanding** [**batch-size** *number*] [**ignore-preferred-server**] **Example:** `Device(config-sg)# load-balance method least-outstanding batch-size 5`	Enables least-outstanding load balancing for a global named server group.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-sg)# end`	Exits server group configuration mode and enters privileged EXEC mode.

Troubleshooting RADIUS Server Load Balancing

After configuring the RADIUS Server Load Balancing feature, you can monitor the idle timer, dead timer, and load balancing server selection or verify the server status by using a manual test command.

SUMMARY STEPS

1. Use the **debug aaa test** command to determine when an idle timer or dead timer has expired, when test packets are sent, the status of the server, or to verify the server state.
2. Use the **debug aaa sg-server selection** command to determine the server that is selected for load balancing.
3. Use the **test aaa group** command to manually verify the RADIUS load-balanced server status.

DETAILED STEPS

Step 1 Use the **debug aaa test** command to determine when an idle timer or dead timer has expired, when test packets are sent, the status of the server, or to verify the server state.

The idle timer is used to check the server status and is updated with or without any incoming requests. Monitoring the idle timer helps to determine if there are nonresponsive servers and to keep the RADIUS server status updated to efficiently utilize available resources. For instance, an updated idle timer would help ensure that incoming requests are sent to servers that are alive.

The dead timer is used either to determine that a server is dead or to update a dead server's status appropriately.

Monitoring server selection helps to determine how often the server selection changes. Server selection is effective in analyzing if there are any bottlenecks, a large number of queued requests, or if only specific servers are processing incoming requests.

The following sample output from the **debug aaa test** command shows when the idle timer expired:

Example:

```
Device# debug aaa test

Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) quarantined.
Jul 16 00:07:01: AAA/SG/TEST: Sending test request(s) to server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Sending 1 Access-Requests, 1 Accounting-Requests in current batch.
Jul 16 00:07:01: AAA/SG/TEST(Req#: 1): Sending test AAA Access-Request.
Jul 16 00:07:01: AAA/SG/TEST(Req#: 1): Sending test AAA Accounting-Request.
Jul 16 00:07:01: AAA/SG/TEST: Obtained Test response from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Obtained Test response from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Necessary responses received from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) marked ALIVE. Idle timer set for 60
sec(s).
Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) removed from quarantine.
```

Step 2 Use the **debug aaa sg-server selection** command to determine the server that is selected for load balancing.

The following sample output from the **debug aaa sg-server selection** command shows five access requests being sent to a server group with a batch size of three:

Example:

```
Device# debug aaa sg-server selection

Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [3] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [2] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [1] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: No more transactions in batch. Obtaining a new server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining a new least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[0] load: 3
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[1] load: 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[2] load: 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Selected Server[1] with load 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [3] transactions remaining in batch.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [2] transactions remaining in batch. Reusing server.
```

Step 3 Use the **test aaa group** command to manually verify the RADIUS load-balanced server status.

The following sample output shows the response from a load-balanced RADIUS server that is alive when the username “test” does not match a user profile. The server is verified alive when it issues an Access-Reject response to an authentication, authorization, and accounting (AAA) packet generated using the **test aaa group** command.

Example:

```
Device# test aaa group SG1 test lab new-code

00:06:07: RADIUS/ENCODE(00000000):Orig. component type = INVALID
00:06:07: RADIUS/ENCODE(00000000): dropping service type, "radius-server attribute 6 on-for-login-auth"
 is off
00:06:07: RADIUS(00000000): Config NAS IP: 192.0.2.4
00:06:07: RADIUS(00000000): sending
00:06:07: RADIUS/ENCODE: Best Local IP-Address 192.0.2.141 for Radius-Server 192.0.2.176
00:06:07: RADIUS(00000000): Send Access-Request to 192.0.2.176:1645 id 1645/1, len 50
00:06:07: RADIUS:  authenticator CA DB F4 9B 7B 66 C8 A9 - D1 99 4E 8E A4 46 99 B4
00:06:07: RADIUS:  User-Password       [2]   18  *
00:06:07: RADIUS:  User-Name           [1]   6   "test"
00:06:07: RADIUS:  NAS-IP-Address      [4]   6   192.0.2.141
00:06:07: RADIUS: Received from id 1645/1 192.0.2.176:1645, Access-Reject, len 44
00:06:07: RADIUS:  authenticator 2F 69 84 3E F0 4E F1 62 - AB B8 75 5B 38 82 49 C3
00:06:07: RADIUS:  Reply-Message       [18]  24
00:06:07: RADIUS:   41 75 74 68 65 6E 74 69 63 61 74 69 6F 6E 20 66  [Authentication f]
00:06:07: RADIUS:   61 69 6C 75 72 65                               [failure]
00:06:07: RADIUS(00000000): Received from id 1645/1
00:06:07: RADIUS/DECODE: Reply-Message fragments, 22, total 22 bytes
```

Configuration Examples for RADIUS Server Load Balancing

Example: Enabling Load Balancing for a Global RADIUS Server Group

The following examples show how to enable load balancing for global RADIUS server groups. These examples are shown in three parts: the current configuration of the RADIUS command output, debug output, and authentication, authorization, and accounting (AAA) server status information. You can use delimiting characters to display relevant parts of the configuration.

The following example shows the relevant RADIUS configuration:

```
Device# show running-config | include radius

aaa authentication ppp default group radius
aaa accounting network default start-stop group radius
radius-server host 192.0.2.238 auth-port 2095 acct-port 2096 key cisco
radius-server host 192.0.2.238 auth-port 2015 acct-port 2016 key cisco
radius-server load-balance method least-outstanding batch-size 5
```

Lines in the current configuration of the preceding RADIUS command output are defined as follows:

- The **aaa authentication ppp** command authenticates all PPP users using RADIUS.
- The **aaa accounting** command enables the sending of all accounting requests to an AAA server when the client is authenticated and then disconnected through use of the **start-stop** keyword.
- The **radius-server host** command defines the IP address of the RADIUS server host with the authorization and accounting ports specified and the authentication and encryption keys identified.
- The **radius-server load-balance** command enables load balancing for global RADIUS server groups with the batch size specified.

The **show debug** sample output below shows the selection of the preferred server and the processing of requests for the configuration:

```
Device# show debug

General OS:
  AAA server group server selection debugging is on
#
<sending 10 pppoe requests>
Device#
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000014):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[0] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Selected Server[0] with load 0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000014):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000015):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000015):Server (192.0.2.238:2095,2096) now being
```

```
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000016):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[3] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000016):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000017):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[2] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000017):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000018):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[1] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000018):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000019):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[0] load:5
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Selected Server[1] with load 0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000019):Server (192.0.2.238:2015,2016) now being
 used as preferred server.
```

The following sample output from the **show aaa servers** command shows the AAA server status for the global RADIUS server group configuration:

The sample output shows the status of two RADIUS servers. Both servers are up and successfully processed in the last 2 minutes:

- Five out of six authentication requests
- Five out of five accounting requests

```
Device# show aaa servers

RADIUS:id 4, priority 1, host 192.0.2.238, auth-port 2095, acct-port 2096
     State:current UP, duration 3175s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 6, timeouts 1
             Response:unexpected 1, server error 0, incorrect 0, time 1841ms
             Transaction:success 5, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 5, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 3303ms
             Transaction:success 5, failure 0
     Elapsed time since counters last cleared:2m
RADIUS:id 5, priority 2, host 192.0.2.238, auth-port 2015, acct-port 2016
     State:current UP, duration 3175s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 6, timeouts 1
             Response:unexpected 1, server error 0, incorrect 0, time 1955ms
             Transaction:success 5, failure 0
```

```
Author:request 0, timeouts 0
        Response:unexpected 0, server error 0, incorrect 0, time 0ms
        Transaction:success 0, failure 0
Account:request 5, timeouts 0
        Response:unexpected 0, server error 0, incorrect 0, time 3247ms
        Transaction:success 5, failure 0
Elapsed time since counters last cleared:2m
```

Example: Server Configuration and Enabling Load Balancing for Global RADIUS Server Group

The following example shows the relevant RADIUS configuration:

```
Device# show running-config | include radius

aaa authentication ppp default group radius
aaa accounting network default start-stop group radius
radius-server host 192.0.2.238 auth-port 2095 acct-port 2096 key cisco
radius-server host 192.0.2.238 auth-port 2015 acct-port 2016 key cisco
radius-server load-balance method least-outstanding batch-size 5
```

Lines in the current configuration of the RADIUS command output above are defined as follows:

- The **aaa authentication ppp** command authenticates all PPP users using RADIUS.
- The **aaa accounting** command enables the sending of all accounting requests to an authentication, authorization, and accounting (AAA) server when the client is authenticated and then disconnected by using the **start-stop** keyword .
- The **radius-server host** command defines the IP address of the RADIUS server host with the authorization and accounting ports specified and the authentication and encryption keys identified.
- The **radius-server load-balance** command enables load balancing for global RADIUS server groups with the batch size specified.

Example: Debug Output for Global RADIUS Server Group

The **debug** command output below shows the selection of the preferred server and the processing of requests for the configuration.

```
Device# show debug

General OS:
  AAA server group server selection debugging is on
#
<sending 10 pppoe requests>
Device#
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000014):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[0] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Selected Server[0] with load 0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000014):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000015):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000015):Server (192.0.2.238:2095,2096) now being
```

```
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000016):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[3] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000016):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000017):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[2] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000017):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000018):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[1] transactions remaining in batch. Reusing
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000018):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000019):No preferred server available.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Server[0] load:5
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:Selected Server[1] with load 0
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:40:32.199:AAA/SG/SERVER_SELECT(00000019):Server (192.0.2.238:2015,2016) now being
 used as preferred server.
```

Example: Server Status Information for Global RADIUS Server Group

The following sample output from the **show aaa server** command shows the AAA server status for the global RADIUS server group configuration:

```
Device# show aaa server

RADIUS:id 4, priority 1, host 192.0.2.238, auth-port 2095, acct-port 2096
     State:current UP, duration 3175s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 6, timeouts 1
             Response:unexpected 1, server error 0, incorrect 0, time 1841ms
             Transaction:success 5, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 5, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 3303ms
             Transaction:success 5, failure 0
     Elapsed time since counters last cleared:2m
RADIUS:id 5, priority 2, host 192.0.2.238, auth-port 2015, acct-port 2016
     State:current UP, duration 3175s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 6, timeouts 1
             Response:unexpected 1, server error 0, incorrect 0, time 1955ms
             Transaction:success 5, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 5, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 3247ms
```

```
            Transaction:success 5, failure 0
     Elapsed time since counters last cleared:2m
```

The sample output shows the status of two RADIUS servers. Both servers are up and successfully processed in the last 2 minutes:

- 5 out of 6 authentication requests
- 5 out of 5 accounting requests

Example: Enabling Load Balancing for a Named RADIUS Server Group

The following examples show load balancing enabled for a named RADIUS server group. These examples are shown in three parts: the current configuration of the RADIUS command output, debug output, and authentication, authorization, and accounting (AAA) server status information.

The following sample output shows the relevant RADIUS configuration:

```
Device# show running-config
.
.
.
aaa group server radius server-group1
 server 192.0.2.238 auth-port 2095 acct-port 2096
 server 192.0.2.238 auth-port 2015 acct-port 2016
 load-balance method least-outstanding batch-size 5
!
aaa authentication ppp default group server-group1
aaa accounting network default start-stop group server-group1
.
.
.
Device(config-sg-radius)# load-balance method least-outstanding batch-size 30
```

The lines in the current configuration of the preceding RADIUS command output are defined as follows:

- The **aaa group server radius** command shows the configuration of a server group with two member servers.
- The **load-balance** command enables load balancing for global RADIUS server groups with the batch size specified.
- The **aaa authentication ppp** command authenticates all PPP users using RADIUS.
- The **aaa accounting** command enables sending of all accounting requests to the AAA server when the client is authenticated and then disconnected using the **start-stop** keyword.

The show debug sample output below shows the selection of the preferred server and the processing of requests for the preceding configuration:

```
Device# show debug

*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002C):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Server[0] load:0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Selected Server[0] with load 0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
```

```
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002C):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002D):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002D):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002E):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[3] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002E):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002F):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[2] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002F):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(00000030):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[1] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(00000030):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000031):No preferred server available.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Server[0] load:5
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Selected Server[1] with load 0
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000031):Server (192.0.2.238:2015,2016) now being
 used as preferred server
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000032):No preferred server available.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
.
.
.
```

The following sample output from the **show aaa servers** command shows the AAA server status for the named RADIUS server group configuration:

The sample output shows the status of two RADIUS servers. Both servers are alive, and no requests have been processed since the counters were cleared 0 minutes ago.

```
Device# show aaa servers

RADIUS:id 8, priority 1, host 192.0.2.238, auth-port 2095, acct-port 2096
     State:current UP, duration 3781s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 0, timeouts 0
```

```
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Elapsed time since counters last cleared:0m
RADIUS:id 9, priority 2, host 192.0.2.238, auth-port 2015, acct-port 2016
     State:current UP, duration 3781s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Elapsed time since counters last cleared:0m
```

Example: Server Configuration and Enabling Load Balancing for Named RADIUS Server Group

The following sample output shows the relevant RADIUS configuration:

```
Device# show running-config
.
.
.
aaa group server radius server-group1
 server 192.0.2.238 auth-port 2095 acct-port 2096
 server 192.0.2.238 auth-port 2015 acct-port 2016
 load-balance method least-outstanding batch-size 5
!
aaa authentication ppp default group server-group1
aaa accounting network default start-stop group server-group1
.
.
.
```

The lines in the current configuration of the RADIUS command output above are defined as follows:

- The **aaa group server radius** command shows the configuration of a server group with two member servers.
- The **load-balance** command enables load balancing for global RADIUS server groups with the batch size specified.
- The **aaa authentication ppp** command authenticates all PPP users using RADIUS.
- The **aaa accounting** command enables sending of all accounting requests to the AAA server when the client is authenticated and then disconnected using the **start-stop** keyword.

Example: Debug Output for Named RADIUS Server Group

The debug sample output below shows the selection of preferred server and processing of requests for the configuration above.

```
Device# show debug

*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002C):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
```

```
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Server[0] load:0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Selected Server[0] with load 0
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002C):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002D):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002D):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002E):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[3] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002E):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002F):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[2] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(0000002F):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(00000030):No preferred server available.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT:[1] transactions remaining in batch. Reusing
server.
*Feb 28 13:51:16.019:AAA/SG/SERVER_SELECT(00000030):Server (192.0.2.238:2095,2096) now being
 used as preferred server
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000031):No preferred server available.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:No more transactions in batch. Obtaining a new
server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining a new least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Server[1] load:0
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Server[0] load:5
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Selected Server[1] with load 0
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:[5] transactions remaining in batch.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000031):Server (192.0.2.238:2015,2016) now being
 used as preferred server
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT(00000032):No preferred server available.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:Obtaining least loaded server.
*Feb 28 13:51:16.023:AAA/SG/SERVER_SELECT:[4] transactions remaining in batch. Reusing
server.
.
.
.
```

Example: Server Status Information for Named RADIUS Server Group

The following sample output from the **show aaa servers** command shows the AAA server status for the named RADIUS server group configuration:

```
Device# show aaa servers

RADIUS:id 8, priority 1, host 192.0.2.238, auth-port 2095, acct-port 2096
     State:current UP, duration 3781s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
```

```
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Elapsed time since counters last cleared:0m
RADIUS:id 9, priority 2, host 192.0.2.238, auth-port 2015, acct-port 2016
     State:current UP, duration 3781s, previous duration 0s
     Dead:total time 0s, count 0
     Quarantined:No
     Authen:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Author:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Account:request 0, timeouts 0
             Response:unexpected 0, server error 0, incorrect 0, time 0ms
             Transaction:success 0, failure 0
     Elapsed time since counters last cleared:0m
```

The sample output shows the status of two RADIUS servers. Both servers are alive, and no requests have been processed since the counters were cleared 0 minutes ago.

Example: Monitoring Idle Timer

The following example shows idle timer and related server state for load balancing enabled for a named RADIUS server group. The current configuration of the RADIUS command output and debug command output are also displayed.

The following sample output shows the relevant RADIUS configuration:

```
Device# show running-config | include radius

aaa group server radius server-group1
radius-server host 192.0.2.238 auth-port 2095 acct-port 2096 test username junk1 idle-time
 1 key cisco
radius-server host 192.0.2.238 auth-port 2015 acct-port 2016 test username junk1 idle-time
 1 key cisco
radius-server load-balance method least-outstanding batch-size 5
```

The lines in the current configuration of the preceding RADIUS command output are defined as follows:

- The **aaa group server radius** command shows the configuration of a server group.
- The **radius-server host** command defines the IP address of the RADIUS server host with authorization and accounting ports specified and the authentication and encryption key identified.
- The **radius-server load-balance** command enables load balancing for the RADIUS server with the batch size specified.

The **show debug** sample output below shows test requests being sent to servers. The response to the test request sent to the server is received, the server is removed from quarantine as appropriate, the server is marked alive, and then the idle timer is reset.

```
Device# show debug

*Feb 28 13:52:20.835:AAA/SG/TEST:Server (192.0.2.238:2015,2016) quarantined.
*Feb 28 13:52:20.835:AAA/SG/TEST:Sending test request(s) to server (192.0.2.238:2015,2016)
*Feb 28 13:52:20.835:AAA/SG/TEST:Sending 1 Access-Requests, 1 Accounting-Requests in current
```

```
 batch.
*Feb 28 13:52:20.835:AAA/SG/TEST(Req#:1):Sending test AAA Access-Request.
*Feb 28 13:52:20.835:AAA/SG/TEST(Req#:1):Sending test AAA Accounting-Request.
*Feb 28 13:52:21.087:AAA/SG/TEST:Obtained Test response from server (192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Obtained Test response from server (192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Necessary responses received from server
(192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Server (192.0.2.238:2015,2016) marked ALIVE. Idle timer
set for 60 secs(s).
*Feb 28 13:52:22.651:AAA/SG/TEST:Server (192.0.2.238:2015,2016) removed from quarantine.
.
.
.
```

Example: Server Configuration and Enabling Load Balancing for Idle Timer Monitoring

The following sample output shows the relevant RADIUS configuration:

```
Device# show running-config | include radius

aaa group server radius server-group1
radius-server host 192.0.2.238 auth-port 2095 acct-port 2096 test username junk1 idle-time
 1 key cisco
radius-server host 192.0.2.238 auth-port 2015 acct-port 2016 test username junk1 idle-time
 1 key cisco
radius-server load-balance method least-outstanding batch-size 5
```

The lines in the current configuration of the RADIUS command output above are defined as follows:

- The **aaa group server radius** command shows the configuration of a server group.
- The **radius-server host** command defines the IP address of the RADIUS server host with authorization and accounting ports specified and the authentication and encryption key identified.
- The **radius-server load-balance** command enables load balancing for the RADIUS server with the batch size specified.

Example: Debug Output for Idle Timer Monitoring

The **debug** command output below shows test requests being sent to servers. The response to the test request sent to the server is received, the server is removed from quarantine as appropriate, marked alive, and then the idle timer is reset.

```
Device# show debug
*Feb 28 13:52:20.835:AAA/SG/TEST:Server (192.0.2.238:2015,2016) quarantined.
*Feb 28 13:52:20.835:AAA/SG/TEST:Sending test request(s) to server (192.0.2.238:2015,2016)
*Feb 28 13:52:20.835:AAA/SG/TEST:Sending 1 Access-Requests, 1 Accounting-Requests in current
 batch.
*Feb 28 13:52:20.835:AAA/SG/TEST(Req#:1):Sending test AAA Access-Request.
*Feb 28 13:52:20.835:AAA/SG/TEST(Req#:1):Sending test AAA Accounting-Request.
*Feb 28 13:52:21.087:AAA/SG/TEST:Obtained Test response from server (192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Obtained Test response from server (192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Necessary responses received from server
(192.0.2.238:2015,2016)
*Feb 28 13:52:22.651:AAA/SG/TEST:Server (192.0.2.238:2015,2016) marked ALIVE. Idle timer
set for 60 secs(s).
*Feb 28 13:52:22.651:AAA/SG/TEST:Server (192.0.2.238:2015,2016) removed from quarantine.
.
.
.
```

Example: Configuring the Preferred Server with the Same Authentication and Authorization Server

The following example shows an authentication server group and an authorization server group that use the same servers 209.165.200.225 and 209.165.200.226. Both server groups have the preferred server flag enabled.

```
aaa group server radius authentication-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
aaa group server radius accounting-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
```

When a preferred server is selected for a session, all transactions for that session will continue to use the original preferred server. The servers 209.165.200.225 and 209.165.200.226 are load balanced based on sessions rather than transactions.

Example: Configuring the Preferred Server with Different Authentication and Authorization Servers

The following example shows an authentication server group that uses servers 209.165.200.225 and 209.165.200.226 and an authorization server group that uses servers 209.165.201.1 and 209.165.201.2. Both server groups have the preferred server flag enabled.

```
aaa group server radius authentication-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
aaa group server radius accounting-group
   server 209.165.201.1 key radkey3
   server 209.165.201.2 key radkey4
```

The authentication server group and the accounting server group do not share any common servers. A preferred server is never found for accounting transactions; therefore, authentication and accounting servers are load-balanced based on transactions. Start and stop records are sent to the same server for a session.

Example: Configuring the Preferred Server with Overlapping Authentication and Authorization Servers

The following example shows an authentication server group that uses servers 209.165.200.225, 209.165.200.226, and 209.165.201.1 and an accounting server group that uses servers 209.165.201.1 and 209.165.201.2. Both server groups have the preferred server flag enabled.

```
aaa group server radius authentication-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
   server 209.165.201.1 key radkey3
aaa group server radius accounting-group
   server 209.165.201.1 key radkey3
   server 209.165.201.2 key radkey4
```

If all servers have equal transaction processing capability, one-third of all authentication transactions are directed toward the server 209.165.201.1. Therefore, one-third of all accounting transactions are also directed toward the server 209.165.201.1. The remaining two-third of accounting transactions are load balanced equally between servers 209.165.201.1 and 209.165.201.2. The server 209.165.201.1 receives fewer authentication transactions because the server 209.165.201.1 has outstanding accounting transactions.

Example: Configuring the Preferred Server with Authentication Servers As a Subset of Authorization Servers

The following example shows an authentication server group that uses servers 209.165.200.225 and 209.165.200.226 and an authorization server group that uses servers 209.165.200.225, 209.165.200.226, and 209.165.201.1. Both server groups have the preferred server flag enabled.

```
aaa group server radius authentication-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
aaa group server radius accounting-group
  server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
   server 209.165.201.1 key radkey3
```

One-half of all authentication transactions are sent to the server 209.165.200.225 and the other half to the server 209.165.200.226. Servers 209.165.200.225 and 209.165.200.226 are preferred servers for authentication and accounting transaction. Therefore, there is an equal distribution of authentication and accounting transactions across servers 209.165.200.225 and 209.165.200.226. The server 209.165.201.1 is relatively unused.

Example: Configuring the Preferred Server with Authentication Servers As a Superset of Authorization Servers

The following example shows an authentication server group that uses servers 209.165.200.225, 209.165.200.226, and 209.165.201.1 and an authorization server group that uses servers 209.165.200.225 and 209.165.200.226. Both server groups have the preferred server flag enabled.

```
aaa group server radius authentication-group
   server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
   server 209.165.201.1 key radkey3
aaa group server radius accounting-group
  server 209.165.200.225 key radkey1
   server 209.165.200.226 key radkey2
```

Initially, one-third of authentication transactions are assigned to each server in the authorization server group. As accounting transactions are generated for more sessions, accounting transactions are sent to servers 209.165.200.225 and 209.165.200.226 because the preferred server flag is on. As servers 209.165.200.225 and 209.165.200.226 begin to process more transactions, authentication transactions will start to be sent to server 209.165.201.1. Transaction requests authenticated by server 209.165.201.1 do not have any preferred server setting and are split between servers 209.165.200.225 and 209.165.200.226, which negates the use of the preferred server flag. This configuration should be used cautiously.

Additional References for RADIUS Server Load Balancing

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
AAA and RADIUS	*Authentication, Authorization, and Accounting Configuration Guide*
AAA server groups and RADIUS configuration	"Configuring RADIUS" module in the *RADIUS Configuration Guide*
Failover retry reorder mode	"RADIUS Server Reorder on Failure" module in the *RADIUS Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

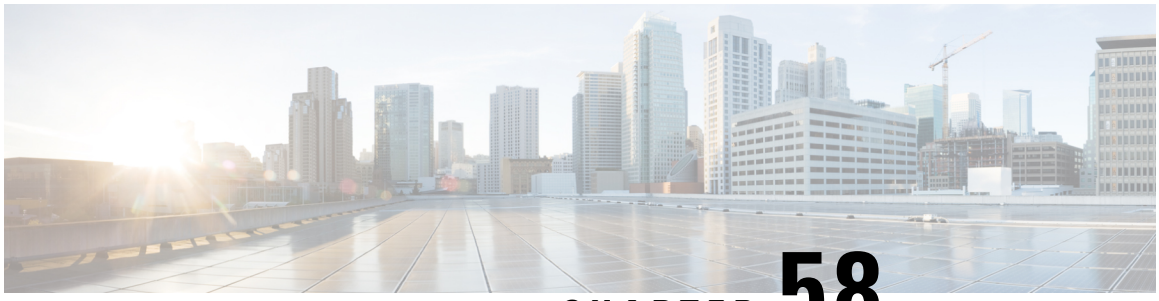
Feature Information for RADIUS Server Load Balancing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 83: Feature Information for RADIUS Server Load Balancing

Feature Name	Releases	Feature Information
RADIUS Server Load Balancing	12.2(28)SB 12.4(11)T 12.2(33)SRC	The RADIUS Server Load Balancing feature distributes authentication, authorization, and accounting (AAA) authentication and accounting transactions across servers in a server group. These servers can then share the transaction load, resulting in faster responses to incoming requests by optimally using available servers. This feature was integrated into Cisco IOS Release 12.2(28)SB. This feature was integrated into Cisco IOS Release 12.4(11)T. This feature was integrated into Cisco IOS Release 12.2(33)SRC. The following commands were introduced or modified: **debug aaa sg-server selection**, **debug aaa test**, **load-balance (server-group)**, **radius-server host**, **radius-server load-balance**, **test aaa group.**
RADIUS Server Load Balancing porting	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 series routers.



CHAPTER 58

RADIUS Server Reorder on Failure

The RADIUS Server Reorder on Failure feature provides for failover to another server in the server group during periods of high load or when server failure occurs. Subsequent to the failure, all RADIUS traffic is directed to the new server. Traffic is switched from the new server to another server in the server group only if the new server also fails. Traffic is not automatically switched back to the first server.

By spreading the RADIUS transactions across multiple servers, authentication and accounting requests are serviced more quickly.

Prerequisites for RADIUS Server Reorder on Failure

- Before you can configure your RADIUS server to perform reorder on failure, you must enable authentication, authorization, and accounting (AAA) by using the **aaa new-model**command.
- You must also have RADIUS configured, for functions such as authentication, accounting, or static route download.

Restrictions for RADIUS Server Reorder on Failure

- An additional 4 bytes of memory is required per server group. However, because most server configurations have only a small number of server groups configured, the additional 4 bytes should have a minimal impact on performance.
- Some RADIUS features within the Cisco IOS XE software set may not be capable of using this feature. If a RADIUS feature cannot use the RADIUS Server Reorder on Failure feature, your server behaves as though the reorder feature is not configured.

Information About RADIUS Server Reorder on Failure

RADIUS Server Failure

If the RADIUS Server Reorder on Failure feature is not configured and server failure occurs:

1. A new RADIUS transaction has to be performed.
2. A RADIUS packet for the transaction is sent to the first server in the group that is not marked dead (as per the configured deadtime) and is retransmitted for the configured number of retransmissions.
3. If all of those retransmits time out (as per the configured timeout), the router transmits the packet to the next nondead server in the list for the configured number of retransmissions.
4. Step 3 is repeated until the specified maximum number of transmissions per transaction have been made. If the end of the list is reached before the maximum number of transmissions has been reached, the router goes back to the beginning of the list and continue from there.

If at any time during this process, a server meets the dead-server detection criteria (not configurable; it varies depending on the version of Cisco IOS XE software being used), the server is marked as dead for the configured deadtime.

How the RADIUS Server Reorder on Failure Feature Works

If you have configured the RADIUS Server Reorder on Failure feature, the decision about which RADIUS server to use as the initial server is as follows:

- The network access server (NAS) maintains the status of “flagged” server, which is the first server to which a transmission is sent.
- After the transmission is sent to the flagged server, the transmission is sent to the flagged server again for the configured number of retransmissions.
- The NAS then sequentially sends the transmission through the list of nondead servers in the server group, starting with the one listed after the flagged server, until the configured transaction maximum tries is reached or until a response is received.
- At boot time, the flagged server is the first server in the server group list as was established using the **radius-server host** command.
- If the flagged server is marked as dead (even if the dead time is zero), the first nondead server listed after the flagged server becomes the flagged server.
- If the flagged server is the last server in the list, and it is marked as dead, the flagged server becomes the first server in the list that is not marked as dead.
- If all servers are marked as dead, the transaction fails, and no change is made to the flagged server.
- If the flagged server is marked as dead, and the dead timer expires, nothing happens.

Note Some types of transmissions (for example, Challenge Handshake Authentication Protocol [CHAP], Microsoft CHAP [MS-CHAP], and Extensible Authentication Protocol [EAP]) require multiple roundtrips to a single server. For these special transactions, the entire sequence of roundtrips to the server are treated as though they were one transmission.

When RADIUS Servers Are Dead

A server can be marked as dead if the criteria in 1 and 2 are met:

1. The server has not responded to at least the configured number of retransmissions as specified by the **radius-server transaction max-tries** command.
2. The server has not responded to any request for at least the configured timeout. The server is marked dead only if both criteria (this and the one listed above) are met. The marking of a server as dead, even if the dead time is zero, is significant for the RADIUS server retry method reorder system.

How to Configure RADIUS Server Reorder on Failure

Configuring a RADIUS Server to Reorder on Failure

Perform this task to configure a server in a server group to direct traffic to another server in the server group when the first server fails.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server retry method reorder**
5. **radius-server retransmit** {*retries*}
6. **radius-server transaction max-tries {** *number* **}**
7. **radius-server host { ***hostname* | *ip-address* **} [key** *string* **]**
8. **radius-server host { ***hostname* | *ip-address* **} [key** *string* **]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**radius-server retry method reorder** **Example:** **Example:** `Router (config)# radius-server retry method reorder`	Specifies the reordering of RADIUS traffic retries among a server group.
Step 5	**radius-server retransmit** {*retries*} **Example:** `Router (config)# radius-server retransmit 1`	Specifies the number of times the Cisco IOS XE software searches the list of RADIUS server hosts before giving up. The *retries* argument is the maximum number of retransmission attempts. The default is 3 attempts.
Step 6	**radius-server transaction max-tries {** *number* **}** **Example:** `Router (config)# radius-server transaction max-tries 3`	Specifies the maximum number of transmissions per transaction that may be retried on a RADIUS server. The *number* argument is the total number of transmissions per transaction. If this command is not configured, the default is eight transmissions. **Note** This command is global across all RADIUS servers for a given transaction.
Step 7	**radius-server host { *hostname* | *ip-address* } [key** *string* **]** **Example:** `Router (config)# radius-server host 10.2.3.4 key radi23`	Specifies a RADIUS server host. **Note** You can also configure a global key for all RADIUS servers that do not have a per-server key configured by issuing the **radius-server key** command.
Step 8	**radius-server host { *hostname* | *ip-address* } [key** *string* **]** **Example:** `Router (config)# radius-server host 10.5.6.7 key rad234`	Specifies a RADIUS server host. **Note** At least two servers must be configured.

Monitoring RADIUS Server Reorder on Failure

To monitor the server-reorder-on-failure process on your router, use the following commands:

SUMMARY STEPS

1. **enable**
2. **debug aaa sg-server selection**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa sg-server selection** **Example:** `Router# debug aaa sg-server selection`	Displays information about why the RADIUS and TACAC+ server group system in the router is choosing a particular server.
Step 3	**debug radius** **Example:** `Router# debug radius`	Displays information about why the router is choosing a particular RADIUS server.

Example

Debug 1

Debug 2

The following two debug outputs display the behavior of the RADIUS Server Reorder on Failure feature:

In the following sample output, the RADIUS Server Reorder on Failure feature is configured. The server retransmits are set to 0 (so each server is tried just one time before failover to the next configured server), and the transmissions per transaction are set to 4 (the transmissions stop on the third failover). The third server in the server group (10.107.164.118) has accepted the transaction on the third transmission (second failover).

```
00:38:35: %SYS-5-CONFIG-I: Configured from console by console
00:38:53: RADIUS/ENCODE(OOOOOOOF) : ask "Username: "
00:38:53: RADIUS/ENCODE (0000000F) : send packet; GET-USER
00:38:58: RADIUS/ENCODE (0000000F) : ask "Password: "
00:38:58: RADIUS/ENCODE(0000000F) : send packet; GET-PASSWORD
00:38:59: RADIUS: AAA Unsupported [152] 4
00:38:59: RADIUS: 7474 [tt]
00:38:59: RADIUS (0000000F) : Storing nasport 2 in rad-db
00:38:59: RADIUS/ENCODE(0000000F) : dropping service type, "radius-server attribute 6
on-for-login-auth" is off
00:38:59: RADIUS (0000000F) : Config NAS IP: 0.0.0.0
00:38:59: RADIUS/ENCODE (0000000F) : acct-session-id: 15
```

```
00:38:59: RADIUS (0000000F) : sending
00:38:59: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 192.1.1.1
00:38:59: RAPIUS(0000000F) : Send Access-Request to 10.10.10.10:1645 id 21645/11, len 78
00:38:59: RADIUS:: authenticator 4481 E6 65 2D 5F 6F OA -lE F5 81 8F 4E 1478 9C
00:38:59: RADIUS: User-Name [1] 7 "username1"
00:38:59: RADIUS: User-Password [2] 18 *
00:38:59: RADIUS: NAS-Port fSl 6 2
00:~8:59: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
00:38:59: RADIUS: Calling-Station-Id [31] 15 "10.19.192.23"
00:39:00: RADIUS: NAS-IP-Address [4] 6 10.0.1.130
00:39:02: RADIUS: Fail-over to (10.2.2.2:1645,1646) for id 21645/11
00:39:02: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 192.2.2.2
00:39:04: RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/11
00:39:04: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 128.107.164.118
00:39:05: RADIUS: Received from id 21645/11 10.107.164.118:1645, Access-Accept, len 26
00:39:05: RADIUS: authenticator 5609 56 F9 64 4E DF 19- F3 A2 DD 73 EE 3F 9826
00:39:05: RADIUS: Service-Type [6] 6 Login [1]
```

In the following sample output, the RADIUS Server Reorder on Failure feature is configured. The server retransmits are set to 0, and the transmissions per transaction are set to 8. In this transaction, the transmission to server 10.10.10.0 has failed on the eighth transmission.

```
00:42:30: RADIUS(00000011): Received from id 21645/13
00:43:34: RADIUS/ENCODE(00000012) : ask "Username: "
00:43:34: RADIUS/ENCODE(00000012) : send packet; GET-USER
00:43:39: RADIUS/ENCODE(00000012) : ask "Password: "
00:43:39: RADIUS/ENCODE(00000012) : send packet; GET-PASSWORD
00:43:40: RADIUS: AAA Unsupported [152] 4
00:43:40: RADIUS: 7474 [tt]
00:43:40: RADIUS(00000012) : Storing nasport 2 in rad-db
00:43:40: RADIUS/ENCODE(00000012): dropping service type, "radius-server attribute 6
on-for-login-auth" is off
00:43:40: RADIUS(00000012) : Co~fig NAS IP: 0.0.0.0
00:43:40: RADIUS/ENCODE(00000012) : acct-session-id: 18
00:43:40: RADIUS(00000012) : sending
00:43:40: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
 00:43:40: RADIUS(00000012) : Send Access-Request to 10.107.164.118:1645 id 21645/14, len
78 00:43:40: RADIUS: authenticator B8 OA 51 3A AF A6 0018 -B3 2E 94 5E 07 OB 2A IF 00:43:40:
 RADIUS: User-Name [1] 7 "username1" 00:43:40: RADIUS: User-Password [2] 18 * 00:43:40:
RADIUS: NAS-Port [5] 6 2
00:43:40: RADIUS: NAS-Port-Type [61] 6 Virtual [5] 00:43:40: RADIUS: Calling-Station-]d
[31] 15 "172.19.192.23" 00:43:40: RADIUS: NAS-IP-Address [4] 6 10.0.1.130
00:43:42: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:42: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.1.1.1 00:43:44:
 RADius: Fail-over to (10.2.2.2:1645,1646) for id 21645/14
00:43:44: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.2.2.2 00:43:46:
 RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/14
00:43:46: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
 00:43:48: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:48: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.1.1.1 00:43:50:
 RADIUS: Fail-over to (10.2.2.2:1645,1646) for id 21645/14
00:43:50: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.2.2.2 00:43:52:
 RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/14
00:43:52: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
 00:43:54: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:54: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.1.1.1 00:43:56:
 RADIUS: No response from (10.10.10.10:1645,1646) for id 21645/14 00:43:56: RADIUS/DECODE:
 parse response no app start; FAIL 00:43:56: RADIUS/DECODE: parse response; FAIL
```

Configuration Examples for RADIUS Server Reorder on Failure

Configuring a RADIUS Server to Reorder on Failure Example

The following configuration example shows that a RADIUS server is configured to reorder on failure. The maximum number of transmissions per transaction that may be retried on the RADIUS server is six.

```
aaa new-model

radius-server retry method reorder

radius-server retransmit 0

radius-server transaction max-tries 6

radius-server host 10.2.3.4 key rad123

radius-server host 10.5.6.7 key rad123
```

Determining Transmission Order When RADIUS Servers Are Dead

If at boot time you have configured the following:

```
Router(config)# aaa new-model
Router(config)# radius-server retry method reorder
Router(config)# radius-server retransmit 0
Router(config)# radius-server transaction max-tries 6
Router(config)# radius-server host 10.2.3.4
Router(config)# radius-server host 10.5.6.7
```

and both servers are down, but not yet marked dead, for the first transaction you would see the transmissions as follows:

```
10.2.3.4
10.5.6.7
10.2.3.4
10.5.6.7
10.2.3.4
10.5.6.7
```

If you configure the reorder as follows:

```
Router(config)# aaa new-model
Router(config)# radius-server retry method reorder
Router(config)# radius-server retransmit 1
Router(config)# radius-server transaction max-tries 3
Router(config)# radius-server host 10.2.3.4
Router(config)# radius-server host 10.4.5.6
```

and both RADIUS servers are not responding to RADIUS packets but are not yet marked dead (as after the NAS boots), the transmissions for the first transaction are as follows:

```
10.2.3.4
10.2.3.4
10.4.5.6
```

Subsequent transactions may be transmitted according to a different pattern. The transmissions depend on whether the criteria for marking one (or both) servers as dead have been met, and as per the server flagging pattern already described.

If you configure the reorder as follows:

```
Router(config)# aaa new-model
Router(config)# radius-server retry method reorder
Router(config)# radius-server retransmit 1
Router(config)# radius-server max-tries-per-transaction 8
Router(config)# radius-server host 10.1.1.1
Router(config)# radius-server host 10.2.2.2
Router(config)# radius-server host 10.3.3.3
Router(config)# radius-server timeout 3
```

And the RADIUS server 10.1.1.1 is not responding to RADIUS packets but is not yet marked as dead, and the remaining two RADIUS servers are live, you see the following:

For the first transaction:

```
10.1.1.1
10.1.1.1
10.2.2.2
```

For any additional transaction initiated for any transmissions before the server is marked as dead:

```
10.1.1.1
10.1.1.1
10.2.2.2
```

For transactions initiated thereafter:

```
10.2.2.2
```

If servers 10.2.2.2 and 10.3.3.3 then go down as well, you see the following transmissions until servers 10.2.2.2 and 10.3.3.3 meet the criteria for being marked as dead:

```
10.2.2.2
10.2.2.2
10.3.3.3
10.3.3.3
10.1.1.1
10.1.1.1
10.2.2.2
10.2.2.2
```

The above is followed by the failure of the transmission and by the next method in the method list being used (if any).

If servers 10.2.2.2 and 10.3.3.3 go down but server 10.1.1.1 comes up at the same time, you see the following:

```
10.2.2.2
10.2.2.2
10.3.3.3
```

```
10.3.3.3
10.1.1.1
```

When servers 10.2.2.2 and 10.3.3.3 are then marked as dead, you see the following:

```
10.1.1.1
```

Additional References

Related Documents

Related Topic	Document Title
RADIUS	"Configuring RADIUS" in the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
AAA and RADIUS commands	*Cisco IOS Security Command Reference*
Enabling AAA	Authentication, Authorization, and Accounting (AAA) section of the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2.
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

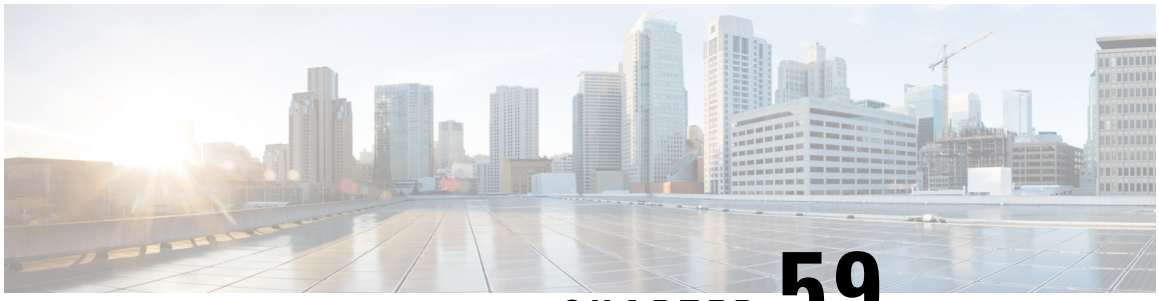
Feature Information for RADIUS Server Reorder on Failure

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 84: Feature Information for RADIUS Server Reorder on Failure

Feature Name	Releases	Feature Information
RADIUS Server Reorder on Failure	Cisco IOS XE Release 2.1	The RADIUS Server Reorder on Failure feature provides for failover to another server in the server group during periods of high load or when server failure occurs. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: **debug aaa sg-server selection, radius-server retry method reorder, radius-server transaction max-tries.**



CHAPTER 59

RADIUS Separate Retransmit Counter for Accounting

The RADIUS: Separate Retransmit Counter for Accounting feature allows users to configure an exponential backoff retransmit. That is, after the normally configured retransmission retries have been used, the router continues trying with an interval that doubles on each retransmission failure until a configured maximum interval is reached. This functionality allows users to retransmit accounting requests for many hours without overloading the RADIUS server when it does come back up.

Restrictions for RADIUS Separate Retransmit Counter for Accounting

The following tasks result in excessive memory consumption on the router:

- Configuring this feature on a router with a high call rate.
- Configuring the **aaa accounting send stop-record authentication failure** command: an accounting record and a RADIUS packet is generated for each user that fails to authenticate while the RADIUS server is down.
- Configuring interim accounting: new accounting records are generated and stored on the router.

Information About RADIUS Separate Retransmit Counter for Accounting

In many environments, a single RADIUS server is used for authentication and accounting. Whenever this server is down for approximately 24 hours, the accounting records of users already on the router are lost after

authentication, authorization, and accounting (AAA) does all the retransmissions. Before the introduction of this feature, the retransmissions could be configured for a maximum of 100 retries and the timeout could be configured for 1,000 seconds. Although these configurations keep the accounting records on the router for 24 hours, a timeout of 1,000 seconds is unreasonable, causing problems when the RADIUS server cannot be reached due to network congestion.

The RADIUS: Separate Retransmit Counter for Accounting feature allows users to configure an exponential backoff retransmit. That is, after the normally configured retransmission retries have been used, the router continues trying with an interval that doubles on each retransmission failure until a configured maximum interval is reached. This functionality allows users to retransmit accounting requests for many hours without overloading the RADIUS server when it does come back up.

This feature can be configured globally (through the **radius-server backoff exponential** command), per server (through the **radius-server host** command), or per group (through the **backoff exponential**command).

Benefits

With this feature, users can extend the time in which the RADIUS client (the router) sends accounting requests to the RADIUS server in the event that the RADIUS server or the connection to the server is down and there is no accounting response confirmation. This functionality enables accounting records to remain on the router for up to 24 hours.

How to Configure RADIUS Separate Retransmit Counter for Accounting

Configuring a Retransmit Counter for Accounting Globally or per RADIUS Host

To configure exponential backoffs of RADIUS retransmits over an extended period of time on a global basis and per RADIUS host, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **radius-server backoff exponential** [**max-delay** *minutes*] [**backoff-retry** *retransmits*
4. Router(config)# **radius-server host** {*hostname* | *ip-address*} [**test username** *user-name*] [**auth-port** *port-number*] [**ignore-auth-port**] [**acct-port** *port-number*] [**ignore-acct-port**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] [**idle-time** *seconds*] [**backoff exponential** {**backoff-retry** *number-of-retransmits* | **key** *encryption-key* | **max-delay** *minutes*}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **radius-server backoff exponential** [**max-delay** *minutes*] [**backoff-retry** *retransmits* **Example:** `Router (config)# radius-server backoff exponential max-delay 60 backoff-retry 32`	Configures the router for exponential backoff retransmit of accounting requests.
Step 4	Router(config)# **radius-server host** {*hostname* | *ip-address*} [**test username** *user-name*] [**auth-port** *port-number*] [**ignore-auth-port**] [**acct-port** *port-number*] [**ignore-acct-port**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] [**idle-time** *seconds*] [**backoff exponential** {**backoff-retry** *number-of-retransmits* | **key** *encryption-key* | **max-delay** *minutes*}] **Example:** `Router (config)# radius-server host 192.0.2.1 test username test1 auth-port 1645 acct-port 1646`	Specifies a RADIUS server host and configures that RADIUS server host for exponential backoff retransmit of accounting requests.

Configuring a Retransmit Counter for Accounting per RADIUS Server Group

To configure exponential backoffs of RADIUS retransmits over an extended period of time per RADIUS server group, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **aaa group server radius** *group-name*
4. Router(config -sg-radius)# **backoff exponential max-delay** *minutes*] [**backoff-retry** *retransmits*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router (config)# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **aaa group server radius** *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server-group RADIUS configuration mode.
Step 4	Router(config -sg-radius)# **backoff exponential max-delay** *minutes*] [**backoff-retry** *retransmits*	Configures the router for exponential backoff retransmit of accounting requests per RADIUS server group.

Verifying Retransmit Configurations

To verify feature functionality, use any of the following EXEC commands:

SUMMARY STEPS

1. **enable**
2. **debug radius**
3. **show accounting**
4. **show radius statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.
Step 3	**show accounting** **Example:** `Router# show accounting`	Displays all active sessions and prints all the accounting records for actively accounted functions.
Step 4	**show radius statistics** **Example:** `Router# show radius statistics`	Displays the RADIUS statistics for accounting packets.

Configuration Examples for RADIUS Separate Retransmit Counter for Accounting

Retransmit Counter for Accounting Comprehensive Configuration Example

The following example shows how to configure your router for exponential backoff retransmit of accounting requests. In this example, an exponential backoff is configured globally (through the **radius-server backoff exponential** command) and for the RADIUS server host "172.107.164.206" (through the **radius-server host** command).

```
aaa new-model
aaa authentication login default group radius
aaa authentication ppp default group radius
aaa authorization exec default group radius
aaa authorization network default group radius
aaa accounting send stop-record authentication failure
aaa accounting update periodic 1
aaa accounting network default start-stop group radius
!
interface BRI1/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation ppp
 no ip mroute-cache
 dialer idle-timeout 0
 dialer-group 1
 isdn switch-type basic-5ess
!
radius-server host 172.107.164.206 auth-port 1645 acct-port 1646 backoff exponential max-delay
 60 backoff-retry 32
radius-server backoff exponential max-delay 60 backoff-retry 32
radius-server retransmit 3
radius-server key rad123
end
```

Per-Server Configuration Example

The following example shows how to enable exponential backoff retransmits on a per-server basis. In this example, assume that the retransmit is configured for 3 retries and the timeout is configured for 5 seconds; that is, the RADIUS request is transmitted 3 times with a delay of 5 seconds. Thereafter, the router continues to retransmit RADIUS requests with a delayed interval that doubles each time until 32 retries have been achieved. The router stops doubling the retransmit intervals after the interval surpasses the configured 60 minutes; it transmits every 60 minutes.

```
radius-server host foo.xyz.com backoff exponential max-delay 60 backoff-retry 32
```

After enabling this command, the retransmits are sent as follows ("t" equals seconds):

```
t = 0 req sent
t = 5 retrans 1
t = 10 retrans 2
t = 15 retrans 3
t = 25 retrans 4
```

```
t = 45 retrans 5
t = 85 retrans 6
t = 165 retrans 7
t = 325 retrans 8
t = 645 retrans 9
t = 1285 retrans 10
t= 2565 retrans 11
t = 5125 retrans 12
t = 8725 retrans 13 (The interval has stabilized to 60 minutes here).
t = 12325 retrans 14 till retransmit 35
```

After all the retransmits are sent, the RADIUS request follows the same path that it would when all the normal retransmits are done.

Additional References

The following sections provide references related to the RADIUS: Separate Retransmit Counter for Accounting.

Related Documents

Related Topic	Document Title
RADIUS and AAA accounting configuration tasks and commands	• " Configuring RADIUS " and " Configuring Accounting " feature modules. • *CiscoOS Security Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

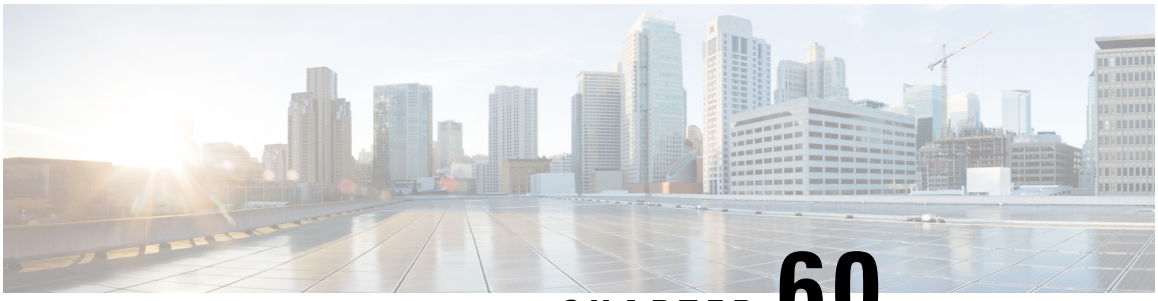
Feature Information for RADIUS Separate Retransmit Counter for Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 85: Feature Information for RADIUS: Separate Retransmit Counter for Accounting

Feature Name	Releases	Feature Information
RADIUS: Separate Retransmit Counter for Accounting	12.2(15)B 12.2(33)SRC	The RADIUS: Separate Retransmit Counter for Accounting feature allows users to configure an exponential backoff retransmit. That is, after the normally configured retransmission retries have been used, the router continues trying with an interval that doubles on each retransmission failure until a configured maximum interval is reached. This functionality allows users to retransmit accounting requests for many hours without overloading the RADIUS server when it does come back up. The following commands were introduced or modified: **backoff exponential**, **radius-server host**, **radius-server backoff exponential**.



CHAPTER 60

RADIUS VC Logging

RADIUS Virtual Circuit (VC) Logging allows the Cisco IOS XE to accurately record the virtual path interface (VPI) and virtual circuit interface (VCI) of an incoming subscriber session.

With RADIUS VC Logging enabled, the RADIUS network access server (NAS)-port field is extended and modified to carry VPI/VCI information. This information is logged in the RADIUS accounting record that was created at session startup.

How to Configure RADIUS VC logging

Configuring the NME Interface IP Address on the NSP

The NAS-IP-Address field in the RADIUS accounting packet contains the IP address of the Network Management Ethernet (NME) port on the Network Service provider (NSP), even if the NME is shut down. If your Network Route Processor (NRP) does not use a DHCP server to obtain an IP address, you must configure a static IP address. Perform the following steps to configure a static combined NME IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface BVI** *bridge-group*
4. **ip address** *address subnet*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface BVI** *bridge-group* **Example:** `Router(config)# interface BVI1`	Selects the combined Bridge-Group Virtual Interface (BVI) NME interface and enters interface configuration mode.
Step 4	**ip address** *address subnet* **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Configures a static IP and subnetwork address.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode.

Configuring the NME IP address

You can use the Gigabit Ethernet port as a separate NME interface instead of the combined NME interface. Perform the following steps to configure the NME IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *number*
4. **ip address** *address mask*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface GigabitEthernet** *number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Selects the NME interface.
Step 4	**ip address** *address mask* **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Configures a static IP and subnetwork address. **Note** You must configure the NME IP address before configuring PVCs on the NRP. Otherwise the NAS-IP-Address field in the RADIUS accounting packet will contain an incorrect IP address.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring RADIUS VC Logging on the NRP

Perform the following steps to configure RADIUS VC logging.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server attribute nas-port format d**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute nas-port format d** **Example:**	Selects the ATM VC (virtual circuit) extended format for the NAS port field.

	Command or Action	Purpose
	`Router(config)# radius-server attribute nas-port format d`	
Step 4	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode.

Verifying the NME Interface IP Address

To verify the NME IP address, enter the **show interface bvi1** or **show interface e0/0/0**EXEC command on the NSP. Check the Internet address statement (indicated with an arrow).

```
Router# show interface bvi1BVI1 is up, line protocol is up
  Hardware is BVI, address is 0010.7ba9.c783 (bia 0000.0000.0000)
    MTU 1500 bytes, BW 10000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type:ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy:fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1540 packets input, 302775 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     545 packets output, 35694 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Verifying RADIUS VC Logging on the NRP

To verify RADIUS VC logging on the RADIUS server, examine a RADIUS accounting packet. If RADIUS VC logging is enabled on the Cisco IOS XE software, the RADIUS accounting packet will appear similar to the following example:

```
Wed Jun 16 13:57:31 1999
NAS-IP-Address = 192.168.100.192
NAS-Port = 268566560
NAS-Port-Type = Virtual
User-Name = "cisco"
Acct-Status-Type = Start
Service-Type = Framed
Acct-Session-Id = "1/0/0/2.32_00000009"
Framed-Protocol = PPP
Framed-IP-Address = 172.16.7.254
Acct-Delay-Time = 0
```

The NAS-Port field shows that RADIUS VC logging is enabled. If this line does not appear in the display, then RADIUS VC logging is not enabled on the Cisco IOS XE software.

The Acct-Session-Id field should also identify the incoming NSP interface and VPI/VCI information, in this format:

```
Acct-Session-Id = "slot/subslot/port/VPI.VCI_acct-session-id"
```

Configuration Examples for RADIUS VC Logging

Example Configuring the NME Interface IP Address on the NSP

The following example shows how to configure a static IP and subnetwork address for the Bridge-Group Virtual Interface:

```
Router> enable
Router# configure terminal
Router(config)# interface BVI1
ip address 209.165.200.225 255.255.255.224
Router(config)# exit
```

Example Configuring the NME IP address

The following example shows how to configure the GigabitEthernet interface:

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config)# exit
```

Example Configuring RADIUS VC Logging on the NRP

The following example shows how to configure the RADIUS VC logging on the NRP:

```
Router> enable
Router# configure terminal
Router(config)# radius-server attribute nas-port format d
Router(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Security Commands List, All Releases*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

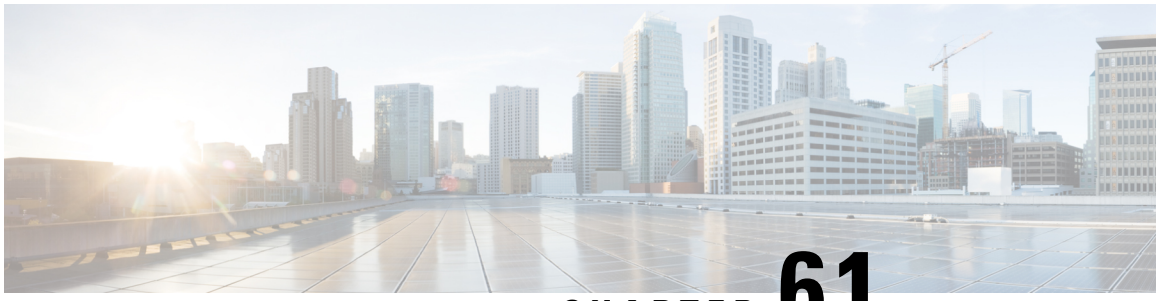
Feature Information for RADIUS VC Logging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 86: Feature Information for Zone-Based Policy Firewall

Feature Name	Releases	Feature Configuration Information
RADIUS VC Logging	Cisco IOS XE Release 3.1S	RADIUS Virtual Circuit (VC) Logging allows the Cisco IOS XE software to accurately record the virtual path interface (VPI) and virtual circuit interface (VCI) of an incoming subscriber session.



CHAPTER 61

RADIUS Centralized Filter Management

The RADIUS Centralized Filter Management feature introduces a filter-server to simplify ACL configuration and management. This filter-server serves as a centralized RADIUS repository and administration point, which users can centrally manage and configure access control list (ACL) filters.

Prerequisites for RADIUS Centralized Filter Management

- You may need to add a dictionary file to your server if it does not support the new RADIUS VSAs. For a sample dictionary and vendors file, see the section “RADIUS Dictionary and Vendors File Example” later in this document.

If you need to add a dictionary file, ensure that your RADIUS server is nonstandard and that it can send the newly introduced VSAs.

- You want to set up RADIUS network authentication so a remote user can dial in and get IP connectivity.

Restrictions for RADIUS Centralized Filter Management

Multiple method lists are not supported in this feature; only a single global filter method list can be configured.

Information About RADIUS Centralized Filter Management

Before the RADIUS Centralized Filter Management feature, wholesale providers (who provide premium charges for customer services such as access control lists [ACLs]) were unable to prevent customers from applying exhaustive ACLs, which could impact router performance and other customers. This feature introduces

a centralized administration point--a filter server--for ACL management. The filter server acts as a centralized RADIUS repository for ACL configuration.

Whether or not the RADIUS server that is used as the filter server is the same server that is used for access authentication, the network access server (NAS) will initiate a second access request to the filter server. If configured, the NAS will use the filter-ID name as the authentication username and the filter server password for the second access request. The RADIUS server will attempt to authenticate the filter-ID name, returning any required filtering configuration in the access-accept response.

Because downloading ACLs is time consuming, a local cache is maintained on the NAS. If an ACL name exists on the local cache, that configuration will be used without consulting the filter server.

Note An appropriately configured cache should minimize delays; however, the first dialin user to require a filter will always experience a longer delay because the ACL configuration is retrieved for the first time.

Cache Management

A global filter cache is maintained on the NAS of recently downloaded ACLs; thus, users no longer have to repeatedly request the same ACL configuration information from a potentially overloaded RADIUS server. Users are required to flush the cache when the following criteria have been met:

- After an entry becomes associated with a newly active call, the idle timer that is associated with that entry will be reset, if configured to do so.
- After the idle-time stamp of an entry expires, the entry will be removed.
- After the global cache of entries reaches a specified maximum number, the entry whose idle-timer is closest to the idle time limit will be removed.

A single timer is responsible for managing all cache entries. The timer is started after the first cache entry is created, and it runs periodically until reboot. The period of the timer will correspond to the minimum granularity offered when configuring cache idle timers, which is one expiration per minute. A single timer prevents users from having to manage individual timers per cache entry.

Note The single timer introduces a lack of precision in timer expiration. There is an average error of approximately 50 percent of the timer granularity. Although decreasing the timer granularity will decrease the average error, the decreased timer granularity will negatively impact performance. Because precise timing is not required for cache management, the error delay should be acceptable.

New Vendor-Specific Attribute Support

This feature introduces support for three new vendor-specific attributes (VSAs), which can be divided into the following two categories:

- User profile extensions
  - Filter-Required (50)--Specifies whether the call should be permitted if the specified filter is not found. If present, this attribute will be applied after any authentication, authorization, and accounting (AAA) filter method-list.

- Pseudo-user profile extensions
  - Cache-Refresh (56)--Specifies whether cache entries should be refreshed each time an entry is referenced by a new session. This attribute corresponds to the **cache refresh** command.
  - Cache-Time (57)--Specifies the idle time out, in minutes, for cache entries. This attribute corresponds to the **cache clear age** command.

Note All RADIUS attributes will override any command-line interface (CLI) configurations.

How to Configure Centralized Filter Management for RADIUS

Configuring the RADIUS ACL Filter Server

To enable the RADIUS ACL filter server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authorization cache filterserver default methodlist[methodlist2...]`	Enables AAA authorization caches and the downloading of an ACL configuration from a RADIUS filter server. • **default** --The default authorization list. • *methodlist* [*methodlist2...*]--One of the keywords listed on the **password** command page.

Configuring the Filter Cache

Follow the steps in this section to configure the AAA filter cache.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **aaa cache filter**
4. Router(config-aaa-filter)# **password 0 7**} *password*
5. Router(config-aaa-filter)# **cache disable**
6. Router(config-aaa-filter)# **cache clear age** *minutes*
7. Router(config-aaa-filter)# **cache refresh**
8. Router(config-aaa-filter)# **cache max** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **aaa cache filter**	Enables filter cache configuration and enters AAA filter configuration mode.
Step 4	Router(config-aaa-filter)# **password 0 7**} *password*	(Optional) Specifies the optional password that is to be used for filter server authentication requests. **0** --Specifies that an unencrypted password will follow. **7** --Specifies that a hidden password will follow. *password* --The unencrypted (clear text) password. **Note** If a password is not specified, the default password ("cisco") is enabled.
Step 5	Router(config-aaa-filter)# **cache disable**	(Optional) Disables the cache.
Step 6	Router(config-aaa-filter)# **cache clear age** *minutes*	(Optional) Specifies, in minutes, when cache entries expire and the cache is cleared. *minutes* --Any value between 0 to 4294967295. **Note** If a time is not specified, the default (1400 minutes [1 day]) is enabled.
Step 7	Router(config-aaa-filter)# **cache refresh**	(Optional) Refreshes a cache entry when a new session begins. This command is enabled by default. To disable this functionality, use the **no cache refresh**command.
Step 8	Router(config-aaa-filter)# **cache max** *number*	(Optional) Limits the absolute number of entries the cache can maintain for a particular server. *number* --The maximum number of entries the cache can contain. Any value between 0 to 4294967295. **Note** If a number is not specified, the default (100 entries) is enabled.

Verifying the Filter Cache

To display the cache status, use the **show aaa cache filterserver** EXEC command. The following is sample output for the **show aaa cache filterserver** command:

```
Router# show aaa cache filterserver
Filter     Server          Age Expires Refresh Access-Control-Lists
-------------------------------------------------------------------------------
aol        10.2.3.4          0    1440     100 ip in icmp drop
                                              ip out icmp drop
                                              ip out forward tcp dstip 1.2.3...
msn        10.3.3.4        N/A   Never       2 ip in tcp drop
msn2       10.4.3.4        N/A   Never       2 ip in tcp drop
vone       10.5.3.4        N/A   Never       0 ip in tcp drop
```

Note The **show aaa cache filterserver** command shows how many times a particular filter has been referenced or refreshed. This function may be used in administration to determine which filters are actually being used.

Troubleshooting Tips

To help troubleshoot your filter cache configurations, use the privileged EXEC **debug aaa cache filterserver** command. To view sample output for the **debug aaa cache filterserver** command, refer to the section "Debug Output Example" later in this document.

Monitoring and Maintaining the Filter Cache

To monitor and maintain filter caches, use at least one of the following EXEC commands:

Command	Purpose
`Router# clear aaa cache filterserver acl [filter-name`	Clears the cache status for a particular filter or all filters.
`Router# show aaa cache filterserver`	Displays the cache status.

Configuration Examples for RADIUS Centralized Filter Management

NAS Configuration Example

The following example shows how to configure the NAS for cache filtering. In this example, the server group "mygroup" in contacted first. If there is no response, the default RADIUS server will then be contacted. If there still is no response, the local filters care contacted. Finally, the call is accepted if the filter cannot be resolved.

```
aaa authorization cache filterserver group mygroup group radius local none
!
aaa group server radius mygroup
 server 10.2.3.4
```

```
 server 10.2.3.5
!
radius-server host 10.1.3.4
!
aaa cache filter
 password mycisco
 no cache refresh
 cache max 100
!
```

RADIUS Server Configuration Example

The following example is a sample RADIUS configuration that is for a remote user “user1” dialing into the NAS:

```
myfilter Password = "cisco"
Service-Type = Outbound,
Ascend:Ascend-Call-Filter = "ip in drop srcip 10.0.0.1/32 dstip 10.0.0.10/32 icmp",
Ascend:Ascend-Call-Filter = "ip in drop srcip 10.0.0.1/32 dstip 10.0.0.10/32 tcp dstport =
 telnet",
Ascend:Ascend-Cache-Refresh = Refresh-No,
Ascend:Ascend-Cache-Time = 15
user1 Password = "cisco"
Service-Type = Framed,
Filter-Id = "myfilter",
Ascend:Ascend-Filter-Required = Filter-Required-Yes,
```

RADIUS Dictionary and Vendors File Example

The following example is a sample RADIUS dictionary file for the new VSAs. In this example, the dictionary file is for a Merit server.

```
dictionary file:
Ascend.attr Ascend-Filter-Required 50 integer (*, 0, NOENCAPS)
Ascend.attr Ascend-Cache-Refresh   56 integer (*, 0, NOENCAPS)
Ascend.attr Ascend-Cache-Time      57 integer (*, 0, NOENCAPS)
Ascend.value Ascend-Cache-Refresh Refresh-No  0
Ascend.value Ascend-Cache-Refresh Refresh-Yes 1
Ascend.value Ascend-Filter-Required  Filter-Required-No  0
Ascend.value Ascend-Filter-Required  Filter-Required-Yes 1
vendors file:
50     50
56     56
57     57
```

Debug Output Example

The following is sample output from the **debug aaa cache filterserver** command:

```
Router# debug aaa cache filterserver

AAA/FLTSV: need "myfilter" (fetch), call 0x612DAC64
AAA/FLTSV: send req, call 0x612DAC50
AAA/FLTSV: method SERVER_GROUP myradius
AAA/FLTSV: recv reply, call 0x612DAC50 (PASS)
AAA/FLTSV: create cache
AAA/FLTSV: add attr "call-inacl"
```

```
AAA/FLTSV: add attr "call-inacl"
AAA/FLTSV: add attr "call-inacl"
AAA/FLTSV: skip attr "filter-cache-refresh"
AAA/FLTSV: skip attr "filter-cache-time"
AAA/CACHE: set "AAA filtserv cache" entry "myfilter" refresh? no
AAA/CACHE: set "AAA filtserv cache" entry "myfilter" cachetime 15
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: PASS call 0x612DAC64
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (0 entries)
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (1 entry)
AAA/CACHE: destroy "AAA filtserv cache" entry "myfilter"
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (0 entries)
```

Additional References

The following sections provide references related to RADIUS Centralized Filter Management.

Related Documents

Related Topic	Document Title
Configuring Authorization	“ Configuring Authorization ” feature module.
Configuring RADIUS	“ Configuring RADIUS ” feature module
Authorization Commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

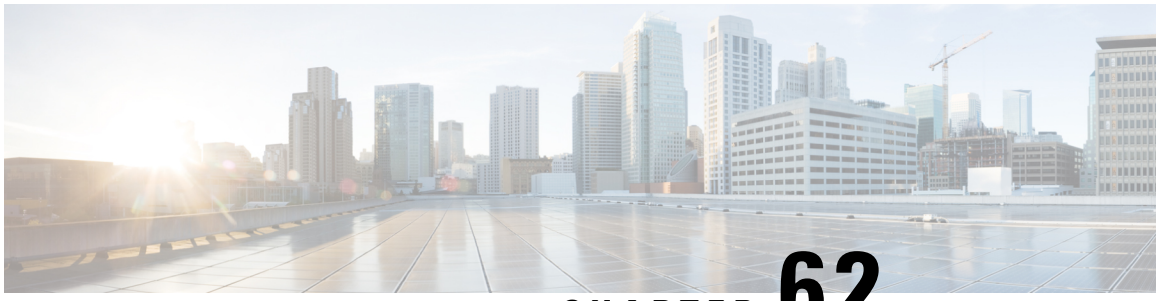
Feature Information for RADIUS Centralized Filter Management

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 87: Feature Information for RADIUS Centralized Filter Management

Feature Name	Releases	Feature Information
RADIUS Centralized Filter Management	Cisco IOS XE Release 3.9S	The RADIUS Centralized Filter Management feature introduces a filter-server to simplify ACL configuration and management. This filter-server serves as a centralized RADIUS repository and administration point, which users can centrally manage and configure access control list (ACL) filters. The following commands were introduced or modified by this feature: **aaa authorization cache filterserver**, **aaa cache filter, cache clear age, cache disable, cache refresh**,**clear aaa cache filterserver acl**, **debug aaa cache filterserver**, **password**, **show aaa cache filterserver.**



CHAPTER 62

RADIUS EAP Support

The RADIUS EAP Support feature makes it possible for users to apply the client authentication methods within PPP (including proprietary authentication), which may not be supported by the network access server (NAS); to be accomplished through the Extensible Authentication Protocol (EAP). Before this feature was introduced, support for various authentication methods for PPP connections required custom vendor-specific configuration and changes to the client and NAS. RADIUS EAP support allows authentication schemes, such as token cards and public key, to strengthen end-user and device authenticated access to their networks.

Prerequisites for RADIUS EAP Support

Before enabling EAP RADIUS on the client, you must perform the following tasks:

- Configure an interface type and enter interface configuration mode by using the **interface** command.
- Configure the interface for PPP encapsulation by using the **encapsulation** command.

For more information on completing these tasks, refer to the " Configuring Asynchronous SLIP and PPP " module.

Restrictions for RADIUS EAP Support

When EAP is running in proxy mode, there may be a significant increase in the authentication time because every packet from the peer must be sent to the RADIUS server and every EAP packet from the RADIUS server must be sent back to the client. Although this extra processing causes delays, you can increase the default authentication timeout value by using the **ppp timeout authentication** command.

Information About RADIUS EAP Support

EAP is an authentication protocol for PPP that supports multiple authentication mechanisms that are negotiated during the authentication phase (instead of the link control protocol [LCP] phase). EAP allows a third-party authentication server to interact with a PPP implementation through a generic interface.

How EAP Works

By default, EAP runs in proxy mode. This means that EAP allows the entire authentication process to be negotiated by the NAS to a back-end server that may reside on or be accessed through a RADIUS server. After EAP is negotiated between the client and the NAS during LCP exchange, all further authentication messages are transparently transmitted between the client and the back-end server. The NAS is no longer directly involved in the authentication process; that is, the NAS works as a proxy, sending EAP messages between the remote peers.

Note EAP can also run in a local mode; the session is authenticated using the Message Digest 5 (MD5) algorithm and obeys the same authentication rules as Challenge Handshake Authentication Protocol (CHAP). To disable proxy mode and authenticate locally, you must use the **ppp eap local** command.

Newly Supported Attributes

The RADIUS EAP Support feature introduces support for the following RADIUS attributes:

Number	IETF Attribute	Description
79	EAP-Message	Encapsulates one fragment of an EAP message, which includes the PPP type, request-id, length, and EAP-type fields.
80	Message Authenticator	Ensures source integrity of the message; all messages that are received with invalid checksums are silently discarded by either end. This attribute contains an HMAC-MD5 checksum of the entire RADIUS request or response message and uses the RADIUS server secret as the key.

How to Configure RADIUS EAP Support

Configuring EAP

Perform this task to configure EAP on an interface configured for PPP encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ppp authentication eap**

4. **ppp eap identity** *string*
5. **ppp eap password** [*number*] *string*
6. **ppp eap local**
7. **ppp eap wait**
8. **ppp eap refuse** [**callin**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ppp authentication eap** **Example:** `Router(config-if)# ppp authentication eap`	Enables EAP as the authentication protocol.
Step 4	**ppp eap identity** *string* **Example:** `Router(config-if)# ppp eap identity user`	(Optional) Specifies the EAP identity when requested by the peer.
Step 5	**ppp eap password** [*number*] *string* **Example:** `Router(config-if)# ppp eap password 7 141B1309`	(Optional) Sets the EAP password for peer authentication. This command should only be configured on the client.
Step 6	**ppp eap local** **Example:** `Router(config-if)# ppp eap local`	(Optional) Authenticates locally instead of using a RADIUS back-end server, which is the default. **Note** This command should only be configured on the NAS.
Step 7	**ppp eap wait** **Example:** `Router(config-if)# ppp eap wait`	(Optional) Waits for the caller to authenticate itself first. By default, the client always authenticates itself before the caller does. **Note** This command should only be configured on the NAS.

	Command or Action	Purpose
Step 8	**ppp eap refuse** [**callin**] **Example:** `Router(config-if)# ppp eap refuse`	(Optional) Refuses to authenticate using EAP. If the **callin** keyword is enabled, only incoming calls are not authenticated. **Note** This command should only be configured on the NAS.

Verifying EAP

To verify EAP configurations on your client or NAS, use at least one of the following commands in privileged EXEC configuration mode:

Command	Purpose
`Router# show users`	Displays information about the active lines on the router.
`Router# show interfaces`	Displays statistics for all interfaces configured on the router or access server.
`Router# show running-config`	Ensures that your configurations appear as part of the running configuration.

Configuration Examples

EAP Local Configuration on Client Example

The following example is a sample configuration for a client configured for EAP:

```
interface Ethernet0/0
 ip address 10.1.1.202 255.255.255.0
 no ip mroute-cache
 half-duplex
!
interface BRI0/0
 ip address 192.168.101.100 255.255.255.0
 encapsulation ppp
 no ip mroute-cache
 dialer map ip 192.168.101.101 56167
 dialer-group 1
 isdn switch-type basic-5ess
 ppp eap identity user
 ppp eap password 7 141B1309
!
!
ip default-gateway 10.1.1.1
ip classless
ip route 192.168.101.101 255.255.255.255 BRI0/0
no ip http server
```

```
!
dialer-list 1 protocol ip permit
```

EAP Proxy Configuration for NAS Example

The following example is a sample configuration for a NAS configured to use EAP proxy:

```
aaa authentication login default group radius
aaa authentication login NOAUTH none
aaa authentication ppp default if-needed group radius
aaa session-id common
enable secret 5 $1$x5D0$cfTL/D8Be.34PgTbdGdgl/
!
username dtw5 password 0 lab
username user password 0 lab
ip subnet-zero
no ip domain-lookup
ip host lab24-boot 172.19.192.254
ip host lb 172.19.192.254
!
isdn switch-type primary-5ess
!
controller T1 3
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0
 ip address 10.1.1.108 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface Serial3:23
 ip address 192.168.101.101 255.255.255.0
 encapsulation ppp
 dialer map ip 192.168.101.100 60213
 dialer-group 1
 isdn switch-type primary-5ess
 isdn T321 0
 ppp authentication eap
 ppp eap password 7 011F0706
!
!
ip default-gateway 10.0.190.1
ip classless
ip route 192.168.101.0 255.255.255.0 Serial3:23
no ip http server
!
dialer-list 1 protocol ip permit
!
radius-server host 10.1.1.201 auth-port 1645 acct-port 1646 key lab
radius-server retransmit 3
call rsvp-sync
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 login authentication NOAUTH
line 1 48
```

```
line aux 0
ine vty 0 4
lpassword lab
```

Additional References

The following sections provide references related to RADIUS EAP Support feature.

Related Documents

Related Topic	Document Title
Configuring PPP Authentication Using AAA	“ Configuring Authentication ” module.
Configuring RADIUS	“ Configuring RADIUS ” module.
PPP Configuration	“ Configuring Asynchronous SLIP and PPP ” module.
Dial Technologies commands	*Cisco IOS Dial Technologies Command Reference*
Security Commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 1938	*A One-Time Password System*
RFC 2869	*RADIUS Extensions*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS EAP Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 88: Feature Information for RADIUS EAP Support

Feature Name	Releases	Feature Information
RADIUS EAP Support	Cisco IOS XE Release 3.9S	The RADIUS EAP Support feature makes it possible for users to apply the client authentication methods within PPP (including proprietary authentication), which may not be supported by the network access server (NAS); to be accomplished through the Extensible Authentication Protocol (EAP). Before this feature was introduced, support for various authentication methods for PPP connections required custom vendor-specific configuration and changes to the client and NAS. RADIUS EAP support allows authentication schemes, such as token cards and public key, to strengthen end-user and device authenticated access to their networks. The following commands were introduced or modified: **ppp authentication**, **ppp eap identity, ppp eap local**, **ppp eap password**, **ppp eap refuse**, **ppp eap wait**.

Glossary

attribute --A RADIUS Internet Engineering Task Force (IETF) attribute is one of the original set of 255 standard attributes that are used to communicate authentication, authorization, and accounting (AAA) information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers that exchange AAA information through IETF attributes must

agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

CHAP --Challenge Handshake Authentication Protocol. Security feature that is supported on lines using PPP encapsulation and prevents unauthorized access. CHAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines whether that user is allowed access.

EAP --Extensible Authentication Protocol. A PPP authentication protocol that supports multiple authentication mechanisms that are negotiated during the authentication phase (instead of the Link Control Protocol [LCP] phase). EAP allows a third-party authentication server to interact with the PPP implementation through a generic interface.

LCP --link control protocol. Protocol that establishes, configures, and tests data-link connections for use by PPP.

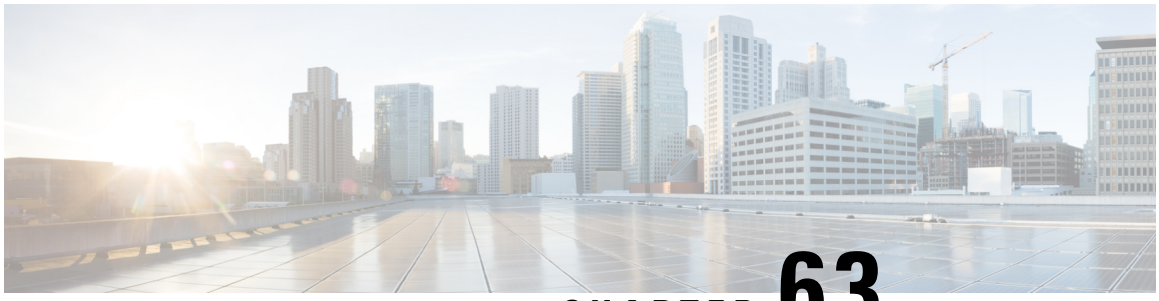
MD5 (HMAC variant) --Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a key hashing for message authentication.

NAS --network access server. A device providing local network access to users across a remote access network such as the public switched telephone network (PSTN).

PAP --Password Authentication Protocol. Authentication protocol that allows PPP peers to authenticate one another. The remote router attempting to connect to the local router is required to send an authentication request. Unlike CHAP, PAP passes the password and host name or username in the clear (unencrypted). PAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines if that user is allowed access. PAP is supported only on PPP lines.

PPP --Point-to-Point Protocol. A protocol that encapsulates network layer protocol information over point-to-point links. PPP is defined in RFC 1661.

RADIUS --Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.



CHAPTER 63

RADIUS Interim Update at Call Connect

The RADIUS Interim Update at Call Connect feature generates an additional accounting record that provides the call connection timestamp for the billing server.

Information About RADIUS Interim Update at Call Connect

When the RADIUS Interim Update at Call Connect feature enabled, Cisco IOS software generates and sends an additional updated interim accounting record to the accounting server when a call leg is connected. A call leg is a distinct segment of a call connection in a voice over IP (VOIP) network that is a logical connection between the router and either a telephony endpoint over a bearer channel, or another endpoint using a session protocol. All attributes (for example, h323-connect-time and backward-call-indicators) available at the time of call connection are sent through this interim updated accounting record.

How to Enable RADIUS Interim Update at Call Connect Feature

Perform the following task to enable the Cisco IOS to generate and send an additional updated interim accounting record to the accounting server when a call leg is connected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **gw-accounting aaa**
5. **aaa accounting update newinfo**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA).
Step 4	**gw-accounting aaa** **Example:** `Router(config)# gw-accounting aaa`	Enables an accounting through the AAA system and sends call detail records (CDRs) to the RADIUS server in the form of vendor-specific attributes (VSAs).
Step 5	**aaa accounting update newinfo** **Example:** `Router(config)# aaa accounting update newinfo`	Enables periodic interim accounting records to be sent to the accounting server whenever there is new accounting information to report relating to the user in question.

Additional References

The following sections provide references related to the RADIUS Interim Update at Call Connect feature.

Related Documents

Related Topic	Document Title
Authentication, Authorization, and Accounting (AAA)	Configuring Authentication , Configuring Authorization , and Configuring Accounting modules.
RADIUS Vendor-Specific Attributes	RADIUS Vendor-Proprietary Attributes module.
Configuring Dynamic Prompts, Customizing Accounting Templates, and Directing AAA Requests for Voice Gateways	*Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T and *Cisco IOS VPDN Configuration Guide* , Release 12.4T.

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2138	*Remote Authentication Dial In User Service (RADIUS)*
RFC 2139	*RADIUS Accounting*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

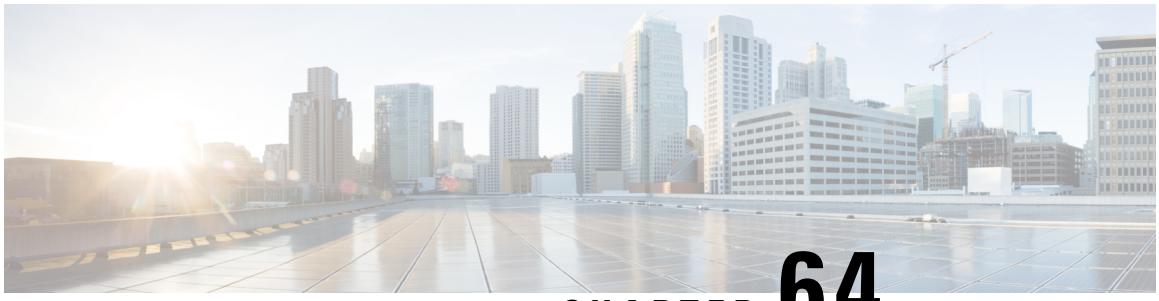
Feature Information for RADIUS Interim Update at Call Connect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 89: Feature Information for RADIUS Interim Update at Call Connect

Feature Name	Releases	Feature Information
RADIUS Interim Update at Call Connect	Cisco IOS XE Release 3.9S	The RADIUS Interim Update at Call Connect feature generates an additional accounting record that provides the call connection timestamp for the billing server. The following commands were introduced or modified: **gw-accounting aaa** and **aaa accounting update**



CHAPTER 64

RADIUS Tunnel Preference for Load Balancing and Fail-Over

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature provides industry-standard load balancing and fail-over functionality for an Layer 2 Tunnel Protocol network server (LNS), rather than requiring the use of a Cisco proprietary Vendor Specific Attribute (VSA). The feature conforms to the tunnel attributes that are to be used in a multivendor network environment as defined in RFC 2868, thereby eliminating interoperability issues among network access servers (NASs) manufactured by different vendors.

Prerequisites

Configuring VPDNs and HGW groups is beyond the scope of this document. See the Related Document section for more information.

Restrictions

The following restrictions and limitations apply to the RADIUS Tunnel Preference for Load Balancing and Fail-Over feature:

- This feature does not support VPDN dial-out networks; it is designed only for dial-in applications.
- The maximum number of LNSs allowed in the network is 1550, which is 50 per tag attribute group and a limit of 31 tags.
- This feature requires a RADIUS server implementation to support RFC 2868.

Information About RADIUS Tunnel Preference for Load Balancing and Fail-Over

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature provides load balancing and fail-over virtual private dialup network (VPDN) home gateway (HGW) groups in a standardized fashion. This feature introduces new software functionality; no new command is associated with this feature.

Industry-Standard Rather Than Proprietary Attributes

Until Cisco IOS Release 12.2(4)T, load balancing and fail-over functionality for a LNS was provided by the Cisco proprietary VSA. In a multivendor network environment, using VSA on a RADIUS server can cause interoperability issues among NASs manufactured by different vendors. Even though some RADIUS server implementations can send VSAs that the requesting NAS can understand, the user still must maintain different VSAs for the same purpose in a single-service profile.

A consensus regarding the tunnel attributes that are to be used in a multivendor network environment is defined in RFC 2868. In RFC 2868, Tunnel-Server-Endpoint, in conjunction with the Tunnel-Medium-Type, specifies the address to which the NAS should initiate a new session. If multiple Tunnel-Server-Endpoint attributes are defined in one tagged attribute group, they are interpreted as equal-cost load-balancing HGWs.

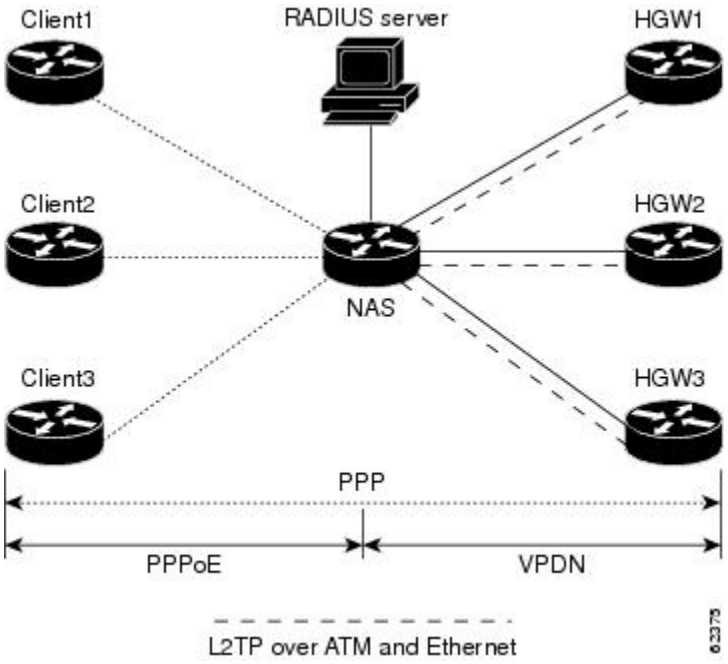
The Tunnel-Preference attribute defined in RFC 2868 can be used as a measure to form load balancing and fail-over HGW groups. When the Tunnel-Preference values of different tagged attribute groups are the same, the Tunnel-Server-Endpoint of those attribute groups is considered to have the same priority unless otherwise specified. When the Tunnel-Preference values of some attribute groups are higher (they have a lower preference) than other attribute groups, their Tunnel-Server-Endpoint attributes will have higher priority values. When an attribute group has a higher priority value, that attribute group will be used for fail-over in case the attribute groups with lower priority values are unavailable for the connections.

Until Cisco IOS Release 12.2(4)T, a specially formatted string would be transported within a Cisco VSA “vpdn:ip-addresses” string to a NAS for the purpose of HGW load balancing and fail-over. For example, 10.0.0.1 10.0.0.2 10.0.0.3/2.0.0.1 2.0.0.2 would be interpreted as IP addresses 10.0.0.1, 10.0.0.2, and 10.0.0.3 for the first group for load balancing. New sessions are projected to these three addresses based on the least-load-first algorithm. This algorithm uses its local knowledge to select an HGW that has the least load to initiate the new session. In this example, the addresses 2.0.0.1 and 2.0.0.2 in the second group have a lower priority and are applicable only when all HGWs specified in the first group fail to respond to the new connection request, thereby making 2.0.0.1 and 2.0.0.2 the fail-over addresses. See the section Configuration Example for RADIUS Tunnel Preference for Load Balancing and Fail-Over, on page 676 for an example of how to configure these fail-over addresses in a RADIUS tunnel profile.

Load Balancing and Fail-Over in a Multivendor Network

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature was designed for large multivendor networks that use VPDN Layer 2 tunnels over WAN links such as ATM and Ethernet, such as the configuration shown in the figure below.

Figure 9: Typical Load Balancing and Fail-Over in a Multivendor Network



In the configuration shown in the figure above, the NAS uses tunnel profiles downloaded from the RADIUS server to establish VPDN Layer 2 tunnels for load balancing and fail-over. The Point-to-Point over Ethernet (PPPoE) protocol is used as the client to generate PPP sessions.

Related Features and Technologies

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature is used in VPDNs. Additionally, familiarity with the following technologies and protocols is recommended:

- ATM
- Ethernet
- L2TP and L2F
- PPP and PPPoE
- RADIUS servers

How RADIUS Tunnel Preference for Load Balancing and Fail-Over is Configured

This feature has no new configuration commands; however, see the next section for an example of how to implement the RADIUS Tunnel Preference for Load Balancing and Fail-Over feature in a RADIUS tunnel profile.

Configuration Example for RADIUS Tunnel Preference for Load Balancing and Fail-Over

The following example shows how to create RADIUS tunnel profiles:

```
net3 Password = "cisco" Service-Type = Outbound
        Tunnel-Type = :0:L2TP,
        Tunnel-Medium-Type = :0:IP,
        Tunnel-Server-Endpoint = :0:"1.1.3.1",
        Tunnel-Assignment-Id = :0:"1",
        Tunnel-Preference = :0:1,
        Tunnel-Password = :0:"welcome"
        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Server-Endpoint = :1:"1.1.5.1",
        Tunnel-Assignment-Id = :1:"1",
        Tunnel-Preference = :1:1,
        Tunnel-Password = :1:"welcome"
        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Server-Endpoint = :2:"1.1.4.1",
        Tunnel-Assignment-Id = :2:"1",
        Tunnel-Preference = :2:1,
        Tunnel-Password = :2:"welcome"
        Tunnel-Type = :3:L2TP,
        Tunnel-Medium-Type = :3:IP,
        Tunnel-Server-Endpoint = :3:"1.1.6.1",
        Tunnel-Assignment-Id = :3:"1",
        Tunnel-Preference = :3:1,
        Tunnel-Password = :3:"welcome"
```

See Information About RADIUS Tunnel Preference for Load Balancing and Fail-Over, on page 674 for more information on how fail-over addresses are selected in these profiles.

Additional References

The following sections provide references related to RADIUS Tunnel Preference for Load Balancing and Fail-Over feature.

Related Documents

Related Topic	Document Title
RADIUS	“ Configuring RADIUS ” module.
RADIUS Attributes	“ RADIUS Attributes Overview and RADIUS IETF Attributes ” module.
Virtual private dialup networks (VPDN) roadmap	*Cisco IOS VPDN Configuration Guide* , Release 15.0.
Dial Technologies	*Cisco IOS Dial Technologies Configuration Guide* , Release 12.4T

Related Topic	Document Title
Broadband Access: PPP and Routed Bridge Encapsulation	*Cisco IOS Broadband Access Aggregation and DSL Configuration Guide* , Release 12.4T

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2868	RADIUS Attributes for Tunnel Protocol Support

Feature Information for RADIUS Tunnel Preference for Load Balancing and Fail-Over

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 90: Feature Information for RADIUS Tunnel Preference for Load Balancing and Fail-Over

Feature Name	Releases	Feature Information
RADIUS Tunnel Preference for Load Balancing and Fail-Over	Cisco IOS XE Release 3.9S	The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature provides industry-standard load balancing and fail-over functionality for an Layer 2 Tunnel Protocol network server (LNS), rather than requiring the use of a Cisco proprietary Vendor Specific Attribute (VSA). The feature conforms to the tunnel attributes that are to be used in a multivendor network environment as defined in RFC 2868, thereby eliminating interoperability issues among network access servers (NASs) manufactured by different vendors.

Glossary

HGW --home gateway. A gateway that terminates Layer 2 tunneling protocols such as L2TP.

home gateway --See HGW.

L2TP --Layer 2 Tunnel Protocol. An Internet Engineering Task Force (IETF) standards track protocol defined in RFC 2661 that provides tunneling of PPP. Based upon the best features of L2F and PPTP, L2TP provides an industry-wide interoperable method of implementing VPDN.

L2TP network server--See LNS.

Layer 2 Tunnel Protocol --See L2TP.

LNS --L2TP network server. A node that acts as one side of an L2TP tunnel endpoint and is a peer to the NAS or L2TP access concentrator (LAC). The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the access server. Analogous to the Layer 2 Forwarding (L2F) HGW.

NAS --network access server. Cisco platform or collection of platforms that interfaces between the packet world (the Internet, for example) and the circuit world (the public switched telephone network, for example).

network access server --See NAS.

Request for Comments --See RFCs.

RFCs --Request for Comments. A series of notes about the Internet collected by the Internet Engineering Task Force (IETF). Started in 1969, the IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture. RFCs define many aspects of computer communication, focusing on networking protocols, procedures, programs, and concepts.

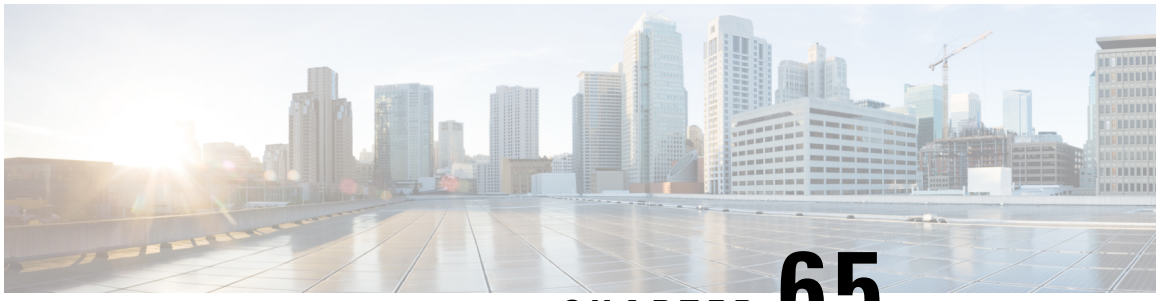
virtual private dialup network --See VPDN.

VPDN --virtual private dialup network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

PART V

RADIUS Attributes



CHAPTER 65

RADIUS Attributes Overview and RADIUS IETF Attributes

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which are stored on the RADIUS program. This chapter lists the RADIUS attributes that are supported.

RADIUS Attributes Overview

IETF Attributes Versus VSAs

RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. The IETF attributes are standard and the attribute data is predefined. All clients and servers that exchange AAA information using IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

RADIUS vendor-specific attributes (VSAs) are derived from a vendor-specific IETF attribute (attribute 26). Attribute 26 allows a vendor to create an additional 255 attributes; that is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26. The newly created attribute is accepted if the user accepts attribute 26.

For more information on VSAs, refer to the chapter “RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values.”

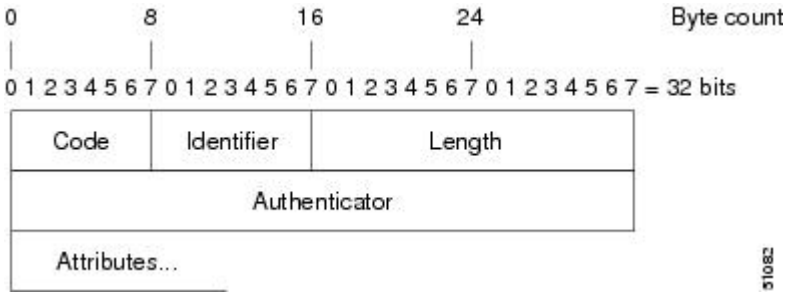
RADIUS Packet Format

The data between a RADIUS server and a RADIUS client is exchanged in RADIUS packets. The data fields are transmitted from left to right.

The figure below shows the fields within a RADIUS packet.

Note For a diagram of VSAs, refer to Figure 1 in the chapter "RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values."

Figure 10: RADIUS Packet Diagram



Each RADIUS packet contains the following information:

- Code—The code field is one octet; it identifies one of the following types of RADIUS packets:
  - Access-Request (1)
  - Access-Accept (2)
  - Access-Reject (3)
  - Accounting-Request (4)
  - Accounting-Response (5)
- Identifier—The identifier field is one octet; it helps the RADIUS server match requests and responses and detect duplicate requests.
- Length—The length field is two octets; it specifies the length of the entire packet.
- Authenticator—The authenticator field is 16 octets. The most significant octet is transmitted first; it is used to authenticate the reply from the RADIUS server. The two types of authenticators are:
  - Request-Authentication: Available in Access-Request and Accounting-Request packets.
  - Response-Authenticator: Available in Access-Accept, Access-Reject, Access-Challenge, and Accounting-Response packets.

RADIUS Packet Types

The following list defines the various types of RADIUS packet types that contain attribute information:

Access-Request—Sent from a client to a RADIUS server. The packet contains information that allows the RADIUS server to determine whether to allow access to a specific network access server (NAS), which will allow access to the user. A user performing authentication must submit an Access-Request packet. After the Access-Request packet is received, the RADIUS server must forward a reply.

Access-Accept—After a RADIUS server receives an Access-Request packet, it must send an Access-Accept packet if all attribute values in the Access-Request packet are acceptable. Access-Accept packets provide the configuration information necessary for the client to provide service to the user.

Access-Reject—After a RADIUS server receives an Access-Request packet, it must send an Access-Reject packet if any of the attribute values are not acceptable.

Access-Challenge—After the RADIUS server receives an Access-Accept packet, it can send the client an Access-Challenge packet, which requires a response. If the client does not know how to respond or if the packets are invalid, the RADIUS server discards the packets. If the client responds to the packet, a new Access-Request packet must be sent with the original Access-Request packet.

Accounting-Request—Sent from a client to a RADIUS accounting server, which provides accounting information. If the RADIUS server successfully records the Accounting-Request packet, it must submit an Accounting Response packet.

Accounting-Response—Sent by the RADIUS accounting server to the client to acknowledge that the Accounting-Request has been received and recorded successfully.

RADIUS Files

Understanding the types of files used by RADIUS is important for communicating AAA information from a client to a server. Each file defines a level of authentication or authorization for the user. The dictionary file defines which attributes the user's NAS can implement, the clients file defines which users are allowed to make requests to the RADIUS server, and the users file defines which user requests the RADIUS server will authenticate based on security and configuration data.

Dictionary File

A dictionary file provides a list of attributes that are dependent on which attributes your NAS supports. However, you can add your own set of attributes to your dictionary for custom solutions. It defines attribute values, so you can interpret attribute output such as parsing requests. A dictionary file contains the following information:

- Name—The ASCII string "name" of the attribute, such as User-Name.
- ID—The numerical "name" of the attribute; for example, User-Name attribute is attribute 1.
- Value type—Each attribute can be specified as one of the following five value types:
  - abinary—0 to 254 octets.
  - date—32-bit value in big-endian order. For example, seconds since 00:00:00 GMT, JAN. 1, 1970.
  - ipaddr—4 octets in network byte order.
  - integer—32-bit value in big-endian order (high byte first).
  - string—0 to 253 octets.

When the data type for a particular attribute is an integer, you can optionally expand the integer to equate to some string. The following sample dictionary includes an integer-based attribute and its corresponding values.

```
# dictionary sample of integer entry
#
ATTRIBUTE       Service-Type            6                     integer
VALUE           Service-Type            Login                    1
VALUE           Service-Type            Framed                   2
VALUE           Service-Type            Callback-Login           3
VALUE           Service-Type            Callback-Framed          4
VALUE           Service-Type            Outbound                 5
VALUE           Service-Type            Administrative           6
VALUE           Service-Type            NAS-Prompt               7
VALUE           Service-Type            Authenticate-Only        8
VALUE           Service-Type            Callback-NAS-Prompt      9
VALUE           Service-Type            Call-Check               10
VALUE           Service-Type            Callback-Administrative  11
```

Clients File

A clients file contains a list of RADIUS clients that are allowed to send authentication and accounting requests to the RADIUS server. To receive authentication, the name and authentication key that the client sends to the server must be an exact match with the data contained in the clients file.

The following is an example of a clients file. The key, as shown in this example, must be the same as the **radius-server key***SomeSecret* command.

```
#Client Name        Key
#----------------  ---------------
10.1.2.3:256        test
nas01               bananas
nas02               MoNkEys
nas07.foo.com       SomeSecret
```

Users File

A RADIUS users file contains an entry for each user that the RADIUS server will authenticate; each entry, which is also known as a user profile, establishes an attribute the user can access.

The first line in any user profile is always a “user access” line; that is, the server must check the attributes on the first line before it can grant access to the user. The first line contains the name of the user, which can be up to 252 characters, followed by authentication information such as the password of the user.

Additional lines, which are associated with the user access line, indicate the attribute reply that is sent to the requesting client or server. The attributes sent in the reply must be defined in the dictionary file. When looking at a user file, note that the data to the left of the equal (=) character is an attribute defined in the dictionary file, and the data to the right of the equal character is the configuration data.

Note A blank line cannot appear anywhere within a user profile.

The following is an example of a RADIUS user profile (Merit Daemon format). In this example, the user name is company.com, the password is user1, and the user can access five tunnel attributes.

```
# This user profile includes RADIUS tunneling attributes
company.com  Password="user1" Service-Type=Outbound
     Tunnel-Type = :1:L2TP
     Tunnel-Medium-Type = :1:IP
     Tunnel-Server-Endpoint = :1:10.0.0.1
     Tunnel-Password = :1:"welcome"
     Tunnel-Assignment-ID = :1:"nas"
```

RADIUS IETF Attributes

Note For RADIUS tunnel attributes, 32 tagged tunnel sets are supported for L2TP.

Supported RADIUS IETF Attributes

Table 1 lists Cisco-supported IETF RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified.

Refer to Table 2 for a description of each listed attribute.

Note Attributes implemented in special (AA) or early development (T) releases are added to the next mainline image.

Table 91: Supported RADIUS IETF Attributes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
1	User-Name	yes	yes	yes	yes	yes	yes	yes	yes
2	User-Password	yes	yes	yes	yes	yes	yes	yes	yes
3	CHAP-Password	yes	yes	yes	yes	yes	yes	yes	yes
4	NAS-IP Address	yes	yes	yes	yes	yes	yes	yes	yes
5	NAS-Port	yes	yes	yes	yes	yes	yes	yes	yes
6	Service-Type	yes	yes	yes	yes	yes	yes	yes	yes
7	Framed-Protocol	yes	yes	yes	yes	yes	yes	yes	yes
8	Framed-IP-Address	yes	yes	yes	yes	yes	yes	yes	yes
9	Framed-IP-Netmask	yes	yes	yes	yes	yes	yes	yes	yes
10	Framed-Routing	yes	yes	yes	yes	yes	yes	yes	yes
11	Filter-Id	yes	yes	yes	yes	yes	yes	yes	yes
12	Framed-MTU	yes	yes	yes	yes	yes	yes	yes	yes
13	Framed-Compression	yes	yes	yes	yes	yes	yes	yes	yes
14	Login-IP-Host	yes	yes	yes	yes	yes	yes	yes	yes
15	Login-Service	yes	yes	yes	yes	yes	yes	yes	yes
16	Login-TCP-Port	yes	yes	yes	yes	yes	yes	yes	yes
18	Reply-Message	yes	yes	yes	yes	yes	yes	yes	yes
19	Callback-Number	no	no	no	no	no	no	yes	yes
20	Callback-ID	no	no	no	no	no	no	no	no
22	Framed-Route	yes	yes	yes	yes	yes	yes	yes	yes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
23	Framed-IPX-Network	no	no	no	no	no	no	no	no
24	State	yes	yes	yes	yes	yes	yes	yes	yes
25	Class	yes	yes	yes	yes	yes	yes	yes	yes
26	Vendor-Specific	yes	yes	yes	yes	yes	yes	yes	yes
27	Session-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
28	Idle-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
29	Termination-Action	no	no	no	no	no	no	no	no
30	Called-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
31	Calling-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
32	NAS-Identifier	no	no	no	no	no	no	no	yes
33	Proxy-State	no	no	no	no	no	no	no	no
34	Login-LAT-Service	yes	yes	yes	yes	yes	yes	yes	yes
35	Login-LAT-Node	no	no	no	no	no	no	no	yes
36	Login-LAT-Group	no	no	no	no	no	no	no	no
37	Framed-AppleTalk-Link	no	no	no	no	no	no	no	no
38	Framed-AppleTalk-Network	no	no	no	no	no	no	no	no
39	Framed-AppleTalk-Zone	no	no	no	no	no	no	no	no
40	Acct-Status-Type	yes	yes	yes	yes	yes	yes	yes	yes
41	Acct-Delay-Time	yes	yes	yes	yes	yes	yes	yes	yes
42	Acct-Input-Octets	yes	yes	yes	yes	yes	yes	yes	yes
43	Acct-Output-Octets	yes	yes	yes	yes	yes	yes	yes	yes
44	Acct-Session-Id	yes	yes	yes	yes	yes	yes	yes	yes
45	Acct-Authentic	yes	yes	yes	yes	yes	yes	yes	yes
46	Acct-Session-Time	yes	yes	yes	yes	yes	yes	yes	yes
47	Acct-Input-Packets	yes	yes	yes	yes	yes	yes	yes	yes
48	Acct-Output-Packets	yes	yes	yes	yes	yes	yes	yes	yes
49	Acct-Terminate-Cause	no	no	no	yes	yes	yes	yes	yes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
50	Acct-Multi-Session-Id	no	yes	yes	yes	yes	yes	yes	yes
51	Acct-Link-Count	no	yes	yes	yes	yes	yes	yes	yes
52	Acct-Input-Gigawords	no	no	no	no	no	no	no	no
53	Acct-Output-Gigawords	no	no	no	no	no	no	no	no
55	Event-Timestamp	no	no	no	no	no	no	no	yes
60	CHAP-Challenge	yes	yes	yes	yes	yes	yes	yes	yes
61	NAS-Port-Type	yes	yes	yes	yes	yes	yes	yes	yes
62	Port-Limit	yes	yes	yes	yes	yes	yes	yes	yes
63	Login-LAT-Port	no	no	no	no	no	no	no	no
64	Tunnel-Type[2]	no	no	no	no	no	no	yes	yes
65	Tunnel-Medium-Type 1	no	no	no	no	no	no	yes	yes
66	Tunnel-Client-Endpoint	no	no	no	no	no	no	yes	yes
67	Tunnel-Server-Endpoint 1	no	no	no	no	no	no	yes	yes
68	Acct-Tunnel-Connection-ID	no	no	no	no	no	no	yes	yes
69	Tunnel-Password 1	no	no	no	no	no	no	yes	yes
70	ARAP-Password	no	no	no	no	no	no	no	no
71	ARAP-Features	no	no	no	no	no	no	no	no
72	ARAP-Zone-Access	no	no	no	no	no	no	no	no
73	ARAP-Security	no	no	no	no	no	no	no	no
74	ARAP-Security-Data	no	no	no	no	no	no	no	no
75	Password-Retry	no	no	no	no	no	no	no	no
76	Prompt	no	no	no	no	no	no	yes	yes
77	Connect-Info	no	no	no	no	no	no	no	yes
78	Configuration-Token	no	no	no	no	no	no	no	no
79	EAP-Message	no	no	no	no	no	no	no	no
80	Message-Authenticator	no	no	no	no	no	no	no	no

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
81	Tunnel-Private-Group-ID	no	no	no	no	no	no	no	no
82	Tunnel-Assignment-ID [2]	no	no	no	no	no	no	yes	yes
83	Tunnel-Preference	no	no	no	no	no	no	no	yes
84	ARAP-Challenge-Response	no	no	no	no	no	no	no	no
85	Acct-Interim-Interval	no	no	no	no	no	no	yes	yes
86	Acct-Tunnel-Packets-Lost	no	no	no	no	no	no	no	no
87	NAS-Port-ID	no	no	no	no	no	no	no	no
88	Framed-Pool	no	no	no	no	no	no	no	no
90	Tunnel-Client-Auth-ID [3]	no	no	no	no	no	no	no	yes
91	Tunnel-Server-Auth-ID	no	no	no	no	no	no	no	yes
200	IETF-Token-Immediate	no	no	no	no	no	no	no	no

[2] This RADIUS attribute complies with the following two draft IETF documents: RFC 2868 RADIUS Attributes for Tunnel Protocol Support and RFC 2867 RADIUS Accounting Modifications for Tunnel Protocol Support.
[3] This RADIUS attribute complies with RFC 2865 and RFC 2868.

Comprehensive List of RADIUS Attribute Descriptions

The table below lists and describes IETF RADIUS attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 92: RADIUS IETF Attributes

Number	IETF Attribute	Description
1	User-Name	Indicates the name of the user being authenticated by the RADIUS server.
2	User-Password	Indicates the user's password or the user's input following an Access-Challenge. Passwords longer than 16 characters are encrypted using RFC 2865 specifications.
3	CHAP-Password	Indicates the response value provided by a PPP Challenge Handshake Authentication Protocol (CHAP) user in response to an Access-Challenge.

Number	IETF Attribute	Description
4	NAS-IP Address	Specifies the IP address of the network access server that is requesting authentication. The default value is 0.0.0.0/0.
5	NAS-Port	Indicates the physical port number of the network access server that is authenticating the user. The NAS-Port value (32 bits) consists of one or two 16-bit values (depending on the setting of the **radius-server extended-portnames** command). Each 16-bit number should be viewed as a 5-digit decimal integer for interpretation as follows: For asynchronous terminal lines, asynchronous network interfaces, and virtual asynchronous interfaces, the value is **00ttt**, where **ttt** is the line number or asynchronous interface unit number. • For ordinary synchronous network interface, the value is **10xxx**. • For channels on a primary rate ISDN interface, the value is **2ppcc** • For channels on a basic rate ISDN interface, the value is **3bb0c**. • For other types of interfaces, the value is **6nnss**.

Number	IETF Attribute	Description
6	Service-Type	Indicates the type of service requested or the type of service to be provided. • In a request: Framed for known PPP or Serial Line Internet Protocol (SLIP) connection. Administrative-user for **enable** command. • In response: Login—Make a connection. Framed--Start SLIP or PPP. Administrative User--Start an EXEC or **enable ok**. Exec User—Start an EXEC session. Service type is indicated by a particular numeric value as follows: • 1: Login • 2: Framed • 3: Callback-Login • 4: Callback-Framed • 5: Outbound • 6: Administrative • 7: NAS-Prompt • 8: Authenticate Only • 9: Callback-NAS-Prompt
7	Framed-Protocol	Indicates the framing to be used for framed access. No other framing is allowed. Framing is indicated by a numeric value as follows: • 1: PPP • 2: SLIP • 3: ARA • 4: Gandalf-proprietary single-link/multilink protocol • 5: Xylogics-proprietary IPX/SLIP

Number	IETF Attribute	Description
8	Framed-IP-Address	Indicates the IP address to be configured for the user, by sending the IP address of a user to the RADIUS server in the access-request. To enable this command, use the **radius-server attribute 8 include-in-access-req** command in global configuration mode.
9	Framed-IP-Netmask	Indicates the IP netmask to be configured for the user when the user is using a device on a network. This attribute value results in a static route being added for Framed-IP-Address with the mask specified.
10	Framed-Routing	Indicates the routing method for the user when the user is using a device on a network. Only "None" and "Send and Listen" values are supported for this attribute. Routing method is indicated by a numeric value as follows: • 0: None • 1: Send routing packets • 2: Listen for routing packets • 3: Send routing packets and listen for routing packets
11	Filter-Id	Indicates the name of the filter list for the user and is formatted as follows: %d, %d.in, or %d.out. This attribute is associated with the most recent service-type command. For login and EXEC, use %d or %d.out as the line access list value from 0 to 199. For Framed service, use %d or %d.out as interface output access list, and %d.in for input access list. The numbers are self-encoding to the protocol to which they refer.
12	Framed-MTU	Indicates the maximum transmission unit (MTU) that can be configured for the user when the MTU is not negotiated by PPP.

Number	IETF Attribute	Description
13	Framed-Compression	Indicates a compression protocol used for the link. This attribute results in a "/compress" being added to the PPP or SLIP autocommand generated during EXEC authorization. This is not implemented for non-EXEC authorization. Compression protocol is indicated by a numeric value as follows: • 0: None • 1: VJ-TCP/IP header compression • 2: IPX header compression
14	Login-IP-Host	Indicates the host to which the user will connect when the Login-Service attribute is included. This begins immediately after login.
15	Login-Service	Indicates the service that should be used to connect the user to the login host. Service is indicated by a numeric value as follows: • 0: Telnet • 1: Rlogin • 2: TCP-Clear • 3: PortMaster • 4: LAT
16	Login-TCP-Port	Defines the TCP port with which the user is to be connected when the Login-Service attribute is also present.
18	Reply-Message	Indicates text that might be displayed to the user using the RADIUS server. You can include this attribute in user files; however, you cannot exceed a maximum of 16 Reply-Message entries per profile.
19	Callback-Number	Defines a dialing string to be used for callback.
20	Callback-ID	Defines the name (consisting of one or more octets) of a place to be called, to be interpreted by the network access server.

Number	IETF Attribute	Description
22	Framed-Route	Provides routing information to be configured for the user on this network access server. The RADIUS RFC format (net/bits [router [metric]]) and the old style dotted mask (net mask [router [metric]]) are supported. If the device field is omitted or 0, the peer IP address is used. Metrics are currently ignored. This attribute is access-request packets.
23	Framed-IPX-Network	Defines the IPX network number configured for the user.
24	State	Allows state information to be maintained between the network access server and the RADIUS server. This attribute is applicable only to CHAP challenges.
25	Class	(Accounting) Arbitrary value that the network access server includes in all accounting packets for this user if supplied by the RADIUS server.

Number	IETF Attribute	Description

26	Vendor-Specific	Allows vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the format:
		
		`protocol : attribute sep value`
		
		"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate AV pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS. For example:
		
		`cisco-avpair= ”ip:addr-pool=first“`
		`cisco-avpair= ”shell:priv-lvl=15“`
		
		The first example causes Cisco's Multiple Named ip address Pools" feature to be activated during IP authorization (during PPP's IPCP address assignment). The second example causes a user logging in from a network access server to have immediate access to EXEC commands.
		
		Table 1 lists supported vendor-specific RADIUS attributes (IETF attribute 26).

27	Session-Timeout	Sets the maximum number of seconds of service to be provided to the user before the session terminates. This attribute value becomes the per-user absolute timeout.
28	Idle-Timeout	Sets the maximum number of consecutive seconds of idle connection allowed to the user before the session terminates. This attribute value becomes the per-user session-timeout.

Number	IETF Attribute	Description
29	Termination-Action	Termination is indicated by a numeric value as follows: • 0: Default • 1: RADIUS request
30	Called-Station-Id	(Accounting) Allows the network access server to send the telephone number the user called as part of the Access-Request packet (using Dialed Number Identification Service [DNIS] or a similar technology). This attribute is only supported on ISDN and modem calls on the Cisco AS5200 if used with PRI.
31	Calling-Station-Id	(Accounting) Allows the network access server to send the telephone number the call came from as part of the Access-Request packet (using Automatic Number Identification or a similar technology). This attribute has the same value as "remote-addr" from TACACS+. This attribute is only supported on ISDN and modem calls on the Cisco AS5200 if used with PRI.
32	NAS-Identifier	String identifying the network access server originating the Access-Request. Use the **radius-server attribute 32 include-in-access-req** global configuration command to send RADIUS attribute 32 in an Access-Request or Accounting-Request. By default, the Fully Qualified Domain Name (FQDN) is sent in the attribute when the format is not specified.
33	Proxy-State	Attribute that can be sent by a proxy server to another server when forwarding Access-Requests; this must be returned unmodified in the Access-Accept, Access-Reject or Access-Challenge and removed by the proxy server before sending the response to the network access server.
34	Login-LAT-Service	Indicates the system with which the user is to be connected by local area transport (LAT). This attribute is only available in the EXEC mode.
35	Login-LAT-Node	Indicates the node with which the user is automatically connected by LAT.

Number	IETF Attribute	Description
36	Login-LAT-Group	Identifies the LAT group codes that the user is authorized to use.
37	Framed-AppleTalk-Link	Indicates the AppleTalk network number that should be used for serial links, which is another AppleTalk device.
38	Framed-AppleTalk- Network	Indicates the AppleTalk network number that the network access server uses to allocate an AppleTalk node for the user.
39	Framed-AppleTalk-Zone	Indicates the AppleTalk Default Zone to be used for the user.
40	Acct-Status-Type	(Accounting) Indicates whether this Accounting-Request marks the beginning of the user service (start) or the end (stop).
41	Acct-Delay-Time	(Accounting) Indicates how many seconds the client has been trying to send a particular record.
42	Acct-Input-Octets	(Accounting) Indicates how many octets have been received from the port over the course of this service being provided.
43	Acct-Output-Octets	(Accounting) Indicates how many octets have been sent to the port in the course of delivering this service.
44	Acct-Session-Id	(Accounting) A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the device is power-cycled or the software is reloaded. To send this attribute in access-request packets, use the **radius-server attribute 44 include-in-access-req** command in global configuration mode.
45	Acct-Authentic	(Accounting) Indicates how the user was authenticated, whether by RADIUS, the network access server itself, or another remote authentication protocol. This attribute is set to “radius” for users authenticated by RADIUS; “remote” for TACACS+ and Kerberos; or “local” for local, enable, line, and if-needed methods. For all other methods, the attribute is omitted.

Number	IETF Attribute	Description
46	Acct-Session-Time	(Accounting) Indicates how long (in seconds) the user has received service.
47	Acct-Input-Packets	(Accounting) Indicates how many packets have been received from the port over the course of this service being provided to a framed user.
48	Acct-Output-Packets	(Accounting) Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.
49	Acct-Terminate-Cause	(Accounting) Reports details on why the connection was terminated. Termination causes are indicated by a numeric value as follows: **1.** User request **2.** Lost carrier **3.** Lost service **4.** Idle timeout **5.** Session timeout **6.** Admin reset **7.** Admin reboot **8.** Port error **9.** NAS error **10.** NAS request **11.** NAS reboot **12.** Port unneeded **13.** Port pre-empted **14.** Port suspended **15.** Service unavailable **16.** Callback **17.** User error **18.** Host request **Note** For attribute 49, Cisco supports values 1 to 6, 8, 9, 12, and 15 to 18.

Number	IETF Attribute	Description
50	Acct-Multi-Session-Id	(Accounting) A unique accounting identifier used to link multiple related sessions in a log file. Each linked session in a multilink session has a unique Acct-Session-Id value, but shares the same Acct-Multi-Session-Id.
51	Acct-Link-Count	(Accounting) Indicates the number of links known in a given multilink session at the time an accounting record is generated. The network access server can include this attribute in any accounting request that might have multiple links.
52	Acct-Input-Gigawords	Indicates how many times the Acct-Input-Octets counter has wrapped around 2^32 over the course of the provided service.
53	Acct-Output-Gigawords	Indicates how many times the Acct-Output-Octets counter has wrapped around 2^32 while delivering service.

Number	IETF Attribute	Description
55	Event-Timestamp	Records the time that the event occurred on the NAS, the timestamp sent in attribute 55 is in seconds since January 1, 1970 00:00 UTC. To send RADIUS attribute 55 in accounting packets, use the **radius-server attribute 55 include-in-acct-req** command. **Note** Before the Event-Timestamp attribute can be sent in accounting packets, you must configure the clock on the network device. (For information on setting the clock on your network device, see the "Performing Basic System Management" section in the "Basic System Management" chapter of *Network Management Configuration Guide*.) To avoid configuring the clock on the network device every time the network device is reloaded, you can enable the **clock calendar-valid** command. (For more information about this command, see the"Setting Time and Calendar Services" section in the "Basic System Management" chapter of *Network Management Configuration Guide*.
60	CHAP-Challenge	Contains the Challenge Handshake Authentication Protocol challenge sent by the network access server to a PPP CHAP user.

Number	IETF Attribute	Description
61	NAS-Port-Type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN-Asynchronous (V.110) • 5: Virtual
62	Port-Limit	Sets the maximum number of ports provided to the user by the NAS.
63	Login-LAT-Port	Defines the port with which the user is to be connected by LAT.
64	Tunnel-Type[4]	Indicates the tunneling protocol(s) used. Cisco software supports one possible value for this attribute: L2TP.
65	Tunnel-Medium-Type1	Indicates the transport medium type used to create a tunnel. This attribute has only one available value for this release: IP. If no value is set for this attribute, IP is used as the default.

Number	IETF Attribute	Description
66	Tunnel-Client-Endpoint	Contains the address of the initiator end of the tunnel. It may be included in both Access-Request and Access-Accept packets to indicate the address from which a new tunnel is to be initiated. If the Tunnel-Client-Endpoint attribute is included in an Access-Request packet, the RADIUS server should take the value as a hint. This attribute should be included in Accounting-Request packets that contain Acct-Status-Type attributes with values of either Start or Stop, in which case it indicates the address from which the tunnel was initiated. This attribute, along with the Tunnel-Server-Endpoint and Acct-Tunnel-Connection-ID attributes, may be used to provide a globally unique method to identify a tunnel for accounting and auditing purposes. An enhancement has been added for the network access server to accept a value of 127.0.0.X for this attribute such that: 127.0.0.0 would indicate that loopback0 IP address has to be used, 127.0.0.1 would indicate that loopback1 IP address has to be used. 127.0.0.X would indicate that loopbackX IP address has to be used for the actual tunnel client endpoint IP address. This enhancement adds scalability across multiple network access servers.
67	Tunnel-Server-Endpoint1	Indicates the address of the server end of the tunnel. The format of this attribute varies depending on the value of Tunnel-Medium-Type. Depending on your release only IP as a tunnel medium type may be supported and the IP address or the host name of LNS is valid for this attribute.
68	Acct-Tunnel-Connection-ID	Indicates the identifier assigned to the tunnel session. This attribute should be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Start, Stop, or any of the values described above. This attribute, along with the Tunnel-Client-Endpoint and Tunnel-Server-Endpoint attributes, may be used to provide a method to uniquely identify a tunnel session for auditing purposes.

Number	IETF Attribute	Description
69	Tunnel-Password1	Defines the password to be used to authenticate to a remote server. This attribute is converted into different AAA attributes based on the value of Tunnel-Type: AAA_ATTR_l2tp_tunnel_pw (L2TP), AAA_ATTR_nas_password (L2F), and AAA_ATTR_gw_password (L2F). By default, all passwords received are encrypted, which can cause authorization failures when a NAS attempts to decrypt a non-encrypted password. To enable attribute 69 to receive non-encrypted passwords, use the **radius-server attribute 69 clear** command in global configuration mode.
70	ARAP-Password	Identifies an Access-Request packet containing a Framed-Protocol of AppleTalk Remote Access Control (ARAP).
71	ARAP-Features	Includes password information that the NAS should send to the user in an ARAP feature flags packet.
72	ARAP-Zone-Access	Indicates how the ARAP zone list for the user should be used.
73	ARAP-Security	Identifies the ARAP Security Module to be used in an Access-Challenge packet.
74	ARAP-Security-Data	Contains the actual security module challenge or response in Access-Challenge and Access-Request packets.
75	Password-Retry	Indicates the number of times a user may attempt authentication before being disconnected.
76	Prompt	Indicates to the NAS whether it should echo the user's response as it is entered or not echo it. (0 = no echo, 1 = echo)
77	Connect-Info	Provides additional call information for modem calls. This attribute is generated in start and stop accounting records.

Number	IETF Attribute	Description
78	Configuration-Token	Indicates the type of user profile to be used. This attribute should be used in large distributed authentication networks based on proxy. It is sent from a RADIUS Proxy Server to a RADIUS Proxy Client in an Access-Accept; it should not be sent to a NAS.
79	EAP-Message	Encapsulates Extended Access Protocol (EAP) packets that allow the NAS to authenticate dial-in users using EAP without having to understand the EAP protocol.
80	Message-Authenticator	Prevents spoofing Access-Requests using CHAP, ARAP, or EAP authentication methods.
81	Tunnel-Private-Group-ID	Indicates the group ID for a particular tunneled session.
82	Tunnel-Assignment-ID1	Indicates to the tunnel initiator the particular tunnel to which a session is assigned.
83	Tunnel-Preference	Indicates the relative preference assigned to each tunnel. This attribute should be included if more than one set of tunneling attributes is returned by the RADIUS server to the tunnel initiator.
84	ARAP-Challenge-Response	Contains the response to the challenge of the dial-in client.
85	Acct-Interim-Interval	Indicates the number of seconds between each interim update in seconds for this specific session. This value can only appear in the Access-Accept message.
86	Acct-Tunnel-Packets-Lost	Indicates the number of packets lost on a given link. This attribute should be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Tunnel-Link-Stop.
87	NAS-Port-ID	Contains a text string which identifies the port of the NAS that is authenticating the user.

Number	IETF Attribute	Description
88	Framed-Pool	Contains the name of an assigned address pool that should be used to assign an address for the user. If a NAS does not support multiple address pools, the NAS should ignore this attribute.
90	Tunnel-Client-Auth-ID	Specifies the name used by the tunnel initiator (also known as the NAS) when authenticating tunnel setup with the tunnel terminator. Supports L2F and L2TP protocols.
91	Tunnel-Server-Auth-ID	Specifies the name used by the tunnel terminator (also known as the Home Gateway) when authenticating tunnel setup with the tunnel initiator. Supports L2F and L2TP protocols.
200	IETF-Token-Immediate	Determines how RADIUS treats passwords received from login-users when their file entry specifies a hand-held security card server. The value for this attribute is indicated by a numeric value as follows: • 0: No—the password is ignored. • 1: Yes—the password is used for authentication.

[4] This RADIUS attribute complies with the following two IETF documents: RFC 2868, RADIUS Attributes for Tunnel Protocol Support and RFC 2867, RADIUS Accounting Modifications for Tunnel Protocol Support .

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

RFCs

RFC	Title
RFC 2865	Remote Authentication Dial In User Service (RADIUS)
RFC 2866	RADIUS Accounting
RFC 2867	RADIUS Accounting Modifications for Tunnel Protocol Support
RFC 2868	RADIUS Attributes for Tunnel Protocol Support
RFC 2869	RADIUS Extensions

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

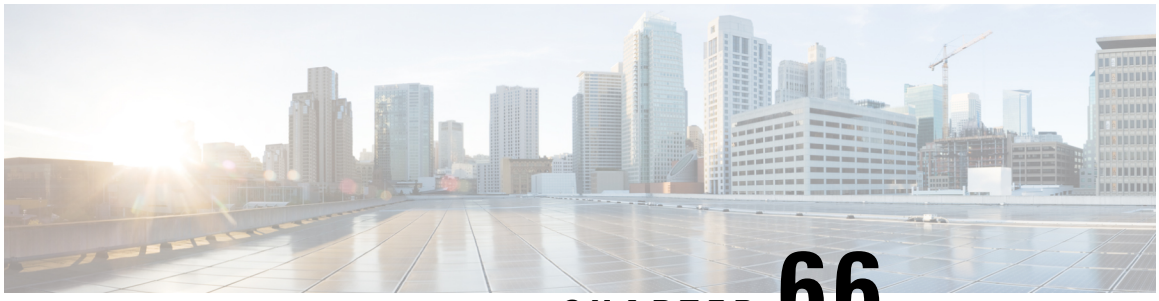
Feature Information for RADIUS Attributes Overview and RADIUS IETF Attributes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 93: Feature Information for RADIUS Attributes Overview and RADIUS IETF Attributes

Feature Name	Releases	Feature Information
RADIUS IETF Attributes	Cisco IOS Release 11.1	This feature was introduced in Cisco IOS Release 11.1.



CHAPTER 66

RADIUS Vendor-Proprietary Attributes

The IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. However, some vendors have extended the RADIUS attribute set for specific applications. This document provides Cisco IOS XE support information for these vendor-proprietary RADIUS attrubutes.

Supported Vendor-Proprietary RADIUS Attributes

The table below lists Cisco-supported vendor-proprietary RADIUS attributes and the Cisco IOS XE release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified. Refer to Refer to Vendor-Proprietary RADIUS Attributes table for a list of descriptions.

Table 94: Supported Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	IOS XE 2.1
17	Change-Password	yes
21	Password-Expiration	yes
68	Tunnel-ID	yes
108	My-Endpoint-Disc-Alias	no
109	My-Name-Alias	no
110	Remote-FW	no
111	Multicast-GLeave-Delay	no
112	CBCP-Enable	no
113	CBCP-Mode	no
114	CBCP-Delay	no

Number	Vendor-Proprietary Attribute	IOS XE 2.1
115	CBCP-Trunk-Group	no
116	Appletalk-Route	no
117	Appletalk-Peer-Mode	no
118	Route-Appletalk	no
119	FCP-Parameter	no
120	Modem-PortNo	no
121	Modem-SlotNo	no
122	Modem-ShelfNo	no
123	Call-Attempt-Limit	no
124	Call-Block-Duration	no
125	Maximum-Call-Duration	no
126	Router-Preference	no
127	Tunneling-Protocol	no
128	Shared-Profile-Enable	no
129	Primary-Home-Agent	no
130	Secondary-Home-Agent	no
131	Dialout-Allowed	no
133	BACP-Enable	no
134	DHCP-Maximum-Leases	no
135	Primary-DNS-Server	yes
136	Secondary-DNS-Server	yes
137	Ascend-Client-Assign-DNS	no
138	User-Acct-Type	no
139	User-Acct-Host	no
140	User-Acct-Port	no
141	User-Acct-Key	no
142	User-Acct-Base	no
143	User-Acct-Time	no

Number	Vendor-Proprietary Attribute	IOS XE 2.1
144	Assign-IP-Client	no
145	Assign-IP-Server	no
146	Assign-IP-Global-Pool	no
147	DHCP-Reply	no
148	DHCP-Pool-Number	no
149	Expect-Callback	no
150	Event-Type	no
151	Ascend-Session-Svr-Key	yes
152	Ascend-Multicast-Rate-Limit	yes
153	IF-Netmask	no
154	h323-Remote-Address	no
155	Ascend-Multicast-Client	yes
156	FR-Circuit-Name	no
157	FR-LinkUp	no
158	FR-Nailed-Grp	no
159	FR-Type	no
160	FR-Link-Mgt	no
161	FR-N391	no
162	FR-DCE-N392	no
163	FR-DTE-N392	no
164	FR-DCE-N393	no
165	FR-DTE-N393	no
166	FR-T391	no
167	FR-T392	no
168	Bridge-Address	no
169	TS-Idle-Limit	no
170	TS-Idle-Mode	no
171	DBA-Monitor	no

Number	Vendor-Proprietary Attribute	IOS XE 2.1
172	Base-Channel-Count	no
173	Minimum-Channels	no
174	IPX-Route	no
175	FT1-Caller	no
176	Ipsec-Backup-Gateway	yes
177	rm-Call-Type	yes
178	Group	no
179	FR-DLCI	no
180	FR-Profile-Name	no
181	Ara-PW	no
182	IPX-Node-Addr	no
183	Home-Agent-IP-Addr	no
184	Home-Agent-Password	no
185	Home-Network-Name	no
186	Home-Agent-UDP-Port	no
187	Multilink-ID	yes
188	Ascend-Num-In-Multilink	yes
189	First-Dest	no
190	Pre-Bytes-In	yes
191	Pre-Bytes-Out	yes
192	Pre-Paks-In	yes
193	Pre-Paks-Out	yes
194	Maximum-Time	yes
195	Disconnect-Cause	yes
196	Connect-Progress	yes
197	Data-Rate	yes
198	PreSession-Time	yes
199	Token-Idle	no

Number	Vendor-Proprietary Attribute	IOS XE 2.1
201	Require-Auth	no
202	Number-Sessions	no
203	Authen-Alias	no
204	Token-Expiry	no
205	Menu-Selector	no
206	Menu-Item	no
207	PW-Warntime	no
208	PW-Lifetime	yes
209	IP-Direct	yes
210	PPP-VJ-Slot-Compression	yes
211	PPP-VJ-1172	no
212	PPP-Async-Map	no
213	Third-Prompt	no
214	Send-Secret	yes
215	Receive-Secret	no
216	IPX-Peer-Mode	no
217	IP-Pool	yes
218	Static-Addr-Pool	yes
219	FR-Direct	no
220	FR-Direct-Profile	no
221	FR-Direct-DLCI	no
222	Handle-IPX	no
223	Netware-Timeout	no
224	IPX-Alias	no
225	Metric	no
226	PRI-Number-Type	no
227	Dial-Number	yes
228	Route-IP	yes

Number	Vendor-Proprietary Attribute	IOS XE 2.1
229	Route-IPX	no
230	Bridge	no
231	Send-Auth	yes
232	Send-Passwd	no
233	Link-Compression	yes
234	Target-Util	yes
235	Maximum-Channels	yes
236	Inc-Channel-Count	no
237	Dec-Channel-Count	no
238	Seconds-of-History	no
239	History-Weigh-Type	no
240	Add-Seconds	no
241	Remove-Seconds	no
242	Data-Filter	yes
243	Call-Filter	no
244	Idle-Limit	yes
245	Preempt-Limit	no
246	Callback	no
247	Data-Service	yes
248	Force-56	yes
249	Billing Number	no
250	Call-By-Call	no
251	Transit-Number	no
252	Host-Info	no
253	PPP-Address	no
254	MPP-Idle-Percent	no
255	Xmit-Rate	yes

Comprehensive List of Vendor-Proprietary RADIUS Attribute Descriptions

The table below lists and describes the known vendor-proprietary RADIUS attributes:

Table 95: Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	Description
17	Change-Password	Specifies a request to change the password of a user.
21	Password-Expiration	Specifies an expiration date for a user's password in the user's file entry.
68	Tunnel-ID	(Ascend 5) Specifies the string assigned by RADIUS for each session using CLID or DNIS tunneling. When accounting is implemented, this value is used for accoutning.
108	My-Endpoint-Disc-Alias	(Ascend 5) No description available.
109	My-Name-Alias	(Ascend 5) No description available.
110	Remote-FW	(Ascend 5) No description available.
111	Multicast-GLeave-Delay	(Ascend 5) No description available.
112	CBCP-Enable	(Ascend 5) No description available.
113	CBCP-Mode	(Ascend 5) No description available.
114	CBCP-Delay	(Ascend 5) No description available.
115	CBCP-Trunk-Group	(Ascend 5) No description available.
116	Appletalk-Route	(Ascend 5) No description available.
117	Appletalk-Peer-Mode	(Ascend 5) No description available.
118	Route-Appletalk	(Ascend 5) No description available.
119	FCP-Parameter	(Ascend 5) No description available.
120	Modem-PortNo	(Ascend 5) No description available.
121	Modem-SlotNo	(Ascend 5) No description available.
122	Modem-ShelfNo	(Ascend 5) No description available.
123	Call-Attempt-Limit	(Ascend 5) No description available.
124	Call-Block-Duration	(Ascend 5) No description available.
125	Maximum-Call-Duration	(Ascend 5) No description available.

Number	Vendor-Proprietary Attribute	Description
126	Router-Preference	(Ascend 5) No description available.
127	Tunneling-Protocol	(Ascend 5) No description available.
128	Shared-Profile-Enable	(Ascend 5) No description available.
129	Primary-Home-Agent	(Ascend 5) No description available.
130	Secondary-Home-Agent	(Ascend 5) No description available.
131	Dialout-Allowed	(Ascend 5) No description available.
133	BACP-Enable	(Ascend 5) No description available.
134	DHCP-Maximum-Leases	(Ascend 5) No description available.
135	Primary-DNS-Server	Identifies a primary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
136	Secondary-DNS-Server	Identifies a secondary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
137	Client-Assign-DNS	No description available.
138	User-Acct-Type	No description available.
139	User-Acct-Host	No description available.
140	User-Acct-Port	No description available.
141	User-Acct-Key	No description available.
142	User-Acct-Base	No description available.
143	User-Acct-Time	No description available.
144	Assign-IP-Client	No description available.
145	Assign-IP-Server	No description available.
146	Assign-IP-Global-Pool	No description available.
147	DHCP-Reply	No description available.
148	DHCP-Pool-Number	No description available.
149	Expect-Callback	No description available.
150	Event-Type	No description available.
151	Session-Svr-Key	No description available.
152	Multicast-Rate-Limit	No description available.
153	IF-Netmask	No description available.

Number	Vendor-Proprietary Attribute	Description
154	Remote-Addr	No description available.
155	Multicast-Client	No description available.
156	FR-Circuit-Name	No description available.
157	FR-LinkUp	No description available.
158	FR-Nailed-Grp	No description available.
159	FR-Type	No description available.
160	FR-Link-Mgt	No description available.
161	FR-N391	No description available.
162	FR-DCE-N392	No description available.
163	FR-DTE-N392	No description available.
164	FR-DCE-N393	No description available.
165	FR-DTE-N393	No description available.
166	FR-T391	No description available.
167	FR-T392	No description available.
168	Bridge-Address	No description available.
169	TS-Idle-Limit	No description available.
170	TS-Idle-Mode	No description available.
171	DBA-Monitor	No description available.
172	Base-Channel-Count	No description available.
173	Minimum-Channels	No description available.
174	IPX-Route	No description available.
175	FT1-Caller	No description available.
176	Backup	No description available.
177	Call-Type	No description available.
178	Group	No description available.
179	FR-DLCI	No description available.
180	FR-Profile-Name	No description available.
181	Ara-PW	No description available.

Number	Vendor-Proprietary Attribute	Description
182	IPX-Node-Addr	No description available.
183	Home-Agent-IP-Addr	Indicates the home agent's IP address (in dotted decimal format) when using Ascend Tunnel Management Protocol (ATMP).
184	Home-Agent-Password	With ATMP, specifies the password that the foreign agent uses to authenticate itself.
185	Home-Network-Name	With ATMP, indicates the name of the connection profile to which the home agent sends all packets.
186	Home-Agent-UDP-Port	Indicates the UDP port number the foreign agent uses to send ATMP messages to the home agent.
187	Multilink-ID	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. The Multilink-ID attribute is sent in authentication-response packets.
188	Num-In-Multilink	Reports the number of sessions remaining in a multilink bundle when the session reported in an accounting-stop packet closes. This attribute applies to sessions that are part of a multilink bundle. The Num-In-Multilink attribute is sent in authentication-response packets and in some accounting-request packets.
189	First-Dest	Records the destination IP address of the first packet received after authentication.
190	Pre-Bytes-In	Records the number of input bytes before authentication. The Pre-Bytes-In attribute is sent in accounting-stop records.
191	Pre-Bytes-Out	Records the number of output bytes before authentication. The Pre-Bytes-Out attribute is sent in accounting-stop records.
192	Pre-Paks-In	Records the number of input packets before authentication. The Pre-Paks-In attribute is sent in accounting-stop records.
193	Pre-Paks-Out	Records the number of output packets before authentication. The Pre-Paks-Out attribute is sent in accounting-stop records.
194	Maximum-Time	Specifies the maximum length of time (in seconds) allowed for any session. After the session reaches the time limit, its connection is dropped.
195	Disconnect-Cause	Specifies the reason a connection was taken offline. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. See the Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values for more information on their meanings.

Number	Vendor-Proprietary Attribute	Description
196	Connect-Progress	Indicates the connection state before the connection is disconnected.
197	Data-Rate	Specifies the average number of bits per second over the course of the connection's lifetime. The Data-Rate attribute is sent in accounting-stop records.
198	PreSession-Time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication. The PreSession-Time attribute is sent in accounting-stop records.
199	Token-Idle	Indicates the maximum amount of time (in minutes) a cached token can remain alive between authentications.
201	Require-Auth	Defines whether additional authentication is required for class that has been CLID authenticated.
202	Number-Sessions	Specifies the number of active sessions (per class) reported to the RADIUS accounting server.
203	Authen-Alias	Defines the RADIUS server's login name during PPP authentication.
204	Token-Expiry	Defines the lifetime of a cached token.
205	Menu-Selector	Defines a string to be used to cue a user to input data.
206	Menu-Item	Specifies a single menu-item for a user-profile. Up to 20 menu items can be assigned per profile.
207	PW-Warntime	(Ascend 5) No description available.
208	PW-Lifetime	Enables you to specify on a per-user basis the number of days that a password is valid.
209	IP-Direct	When you include this attribute in a user's file entry, a framed route is installed to the routing and bridging tables. **Note** Packet routing is dependent upon the entire table, not just this newly installed entry. The inclusion of this attribute does not guarantee that all packets should be sent to the specified IP address; thus, this attribute is not fully supported. These attribute limitations occur because the Cisco router cannot bypass all internal routing and bridging tables and send packets to a specified IP address.
210	PPP-VJ-Slot-Comp	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.
211	PPP-VJ-1172	Instructs PPP to use the 0x0037 value for VJ compression.

Number	Vendor-Proprietary Attribute	Description
212	PPP-Async-Map	Gives the Cisco router the asynchronous control character map for the PPP session. The specified control characters are passed through the PPP link as data and used by applications running over the link.
213	Third-Prompt	Defines a third prompt (after username and password) for additional user input.
214	Send-Secret	Enables an encrypted password to be used in place of a regular password in outdial profiles.
215	Receive-Secret	Enables an encrypted password to be verified by the RADIUS server.
216	IPX-Peer-Mode	(Ascend 5) No description available.
217	IP-Pool-Definition	Defines a pool of addresses using the following format: X a.b.c Z; where X is the pool index number, a.b.c is the pool's starting IP address, and Z is the number of IP addresses in the pool. For example, 3 10.0.0.1 5 allocates 10.0.0.1 through 10.0.0.5 for dynamic assignment.
218	Assign-IP-Pool	Tells the router to assign the user and IP address from the IP pool.
219	FR-Direct	Defines whether the connection profile operates in Frame Relay redirect mode.
220	FR-Direct-Profile	Defines the name of the Frame Relay profile carrying this connection to the Frame Relay switch.
221	FR-Direct-DLCI	Indicates the DLCI carrying this connection to the Frame Relay switch.
222	Handle-IPX	Indicates how NCP watchdog requests will be handled.
223	Netware-Timeout	Defines, in minutes, how long the RADIUS server responds to NCP watchdog packets.
224	IPX-Alias	Allows you to define an alias for IPX routers requiring numbered interfaces.
225	Metric	No description available.
226	PRI-Number-Type	No description available.
227	Dial-Number	Defines the number to dial.
228	Route-IP	Indicates whether IP routing is allowed for the user's file entry.
229	Route-IPX	Allows you to enable IPX routing.
230	Bridge	No description available.
231	Send-Auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.

Number	Vendor-Proprietary Attribute	Description
232	Send-Passwd	Enables the RADIUS server to specify the password that is sent to the remote end of a connection on outgoing calls.
233	Link-Compression	Defines whether to turn on or turn off "stac" compression over a PPP link. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac
234	Target-Util	Specifies the load-threshold percentage value for bringing up an additional channel when PPP multilink is defined.
235	Maximum-Channels	Specifies allowed/allocatable maximum number of channels.
236	Inc-Channel-Count	No description available.
237	Dec-Channel-Count	No description available.
238	Seconds-of-History	No description available.
239	History-Weigh-Type	No description available.
240	Add-Seconds	No description available.
241	Remove-Seconds	No description available.
242	Data-Filter	Defines per-user IP data filters. These filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile. Filter entries are applied on a first-match basis; therefore, the order in which filter entries are entered is important.
243	Call-Filter	Defines per-user IP data filters. On a Cisco router, this attribute is identical to the Data-Filter attribute.
244	Idle-Limit	Specifies the maximum time (in seconds) that any session can be idle. When the session reaches the idle time limit, its connection is dropped.
245	Preempt-Limit	No description available.
246	Callback	Allows you to enable or disable callback.
247	Data-Svc	No description available.
248	Force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.

Number	Vendor-Proprietary Attribute	Description
249	Billing Number	No description available.
250	Call-By-Call	No description available.
251	Transit-Number	No description available.
252	Host-Info	No description available.
253	PPP-Address	Indicates the IP address reported to the calling unit during PPP IPCP negotiations.
254	MPP-Idle-Percent	No description available.
255	Xmit-Rate	(Ascend 5) No description available.

See the Configuring RADIUS feature module for more information on vendor-propritary RADIUS attributes.

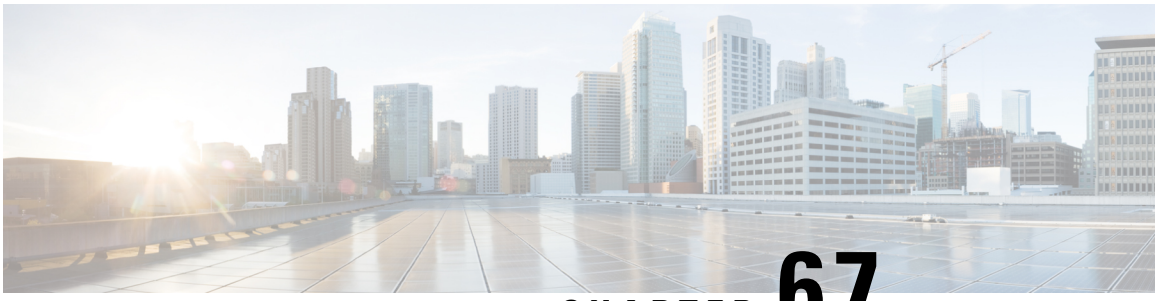
Feature Information for RADIUS Vendor-Proprietary Attributes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 96: Feature Information for RADIUS Vendor-Proprietary Attributes

Feature Name	Releases	Feature Information
RADIUS Vendor-Proprietary Attributes	Cisco IOS XE Release 2.1	The IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. However, some vendors have extended the RADIUS attribute set for specific applications. This document provides Cisco IOS XE support information for these vendor-proprietary RADIUS attrubutes. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 67

RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

Information About RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

```
cisco-avpair= ”ip:addr-pool*first“
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

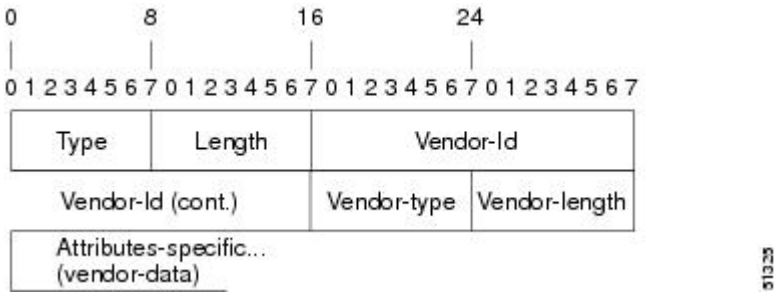
```
cisco-avpair= ”shell:priv-lvl=15“
```

Attribute 26 contains the following three elements:

- Type
- Length
- String (also known as data)
  - Vendor-Id
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

The figure below shows the packet format for a VSA encapsulated “behind” attribute 26.

Figure 11: VSA Encapsulated Behind Attribute 26



Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

The table below describes significant fields listed in the Vendor-Specific RADIUS IETF Attributes table (second table below), which lists supported vendor-specific RADIUS attributes (IETF attribute 26).

Table 97: Vendor-Specific Attributes Table Field Descriptions

Field	Description
Number	All attributes listed in the following table are extensions of IETF attribute 26.
Vendor-Specific Command Codes	A defined code used to identify a particular vendor. Code 9 defines Cisco VSAs, 311 defines Microsoft VSAs, and 529 defines Ascend VSAs.
Sub-Type Number	The attribute ID number. This number is much like the ID numbers of IETF attributes, except it is a “second layer” ID number encapsulated behind attribute 26.
Attribute	The ASCII string name of the attribute.

Field	Description
Description	Description of the attribute.

Table 98: Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548)
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template.
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	tunnel-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no.
H323 Attributes				
26	9	23	Remote-Gateway-ID (h323-remote-address)	Indicates the IP address of the remote gateway.
26	9	24	Connection-ID (h323-conf-id)	Identifies the conference ID.
26	9	25	Setup-Time (h323-setup-time)	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	Call-Origin (h323-call-origin)	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	Call-Type (h323-call-type)	Indicates call leg type. Possible values are **telephony** and **VoIP**.
26	9	28	Connect-Time (h323-connect-time)	Indicates the connection time for this call leg in UTC.
26	9	29	Disconnect-Time (h323-disconnect-time)	Indicates the time this call leg was disconnected in UTC.
26	9	30	Disconnect-Cause (h323-disconnect-caus)e	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	Voice-Quality (h323-voice-quality)	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	Gateway-ID (h323-gw-id)	Indicates the name of the underlying gateway.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
Miscellaneous Attributes				
26	9	2	Cisco-NAS-Port	Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting. To specify additional NAS-Port information in the form an Attribute-Value Pair (AVPair) string, use the radius-server vsa send global configuration command. **Note** This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.
26	9	1	min-links	Sets the minimum number of links for MLP.
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	client-mac-address	Contains the MAC address of the PPPoE client. **Note** This attribute is applicable only to PPP over Ethernet (PPPoE) or to PPP over ATM (PPPoA).

See “Configuring Router to Use Vendor-Specific RADIUS Attributes” section of t he Configuring RADIUS feature module for more information on configuring your NAS to recognize and use VSAs.

RADIUS Disconnect-Cause Attribute Values

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

The table below lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

Note The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

Table 99: Disconnect-Cause Attribute Values

Cause Code	Value	Description
2	Unknown	Reason unknown.
4	CLID-Authentication-Failure	Failure to authenticate number of the calling-party.
10	No-Carrier	No carrier detected. **Note** Codes 10, 11, and 12 can be sent if there is a disconnection during initial modem connection.
11	Lost-Carrier	Loss of carrier.
12	No-Detected-Result-Codes	Failure to detect modem result codes.
20	User-Ends-Session	User terminates a session. **Note** Codes 20, 22, 23, 24, 25, 26, 27, and 28 apply to EXEC sessions.
21	Idle-Timeout	Timeout waiting for user input. Codes 21, 100, 101, 102, and 120 apply to all session types.
22	Exit-Telnet-Session	Disconnect due to exiting Telnet session.

Cause Code	Value	Description
23	No-Remote-IP-Addr	Could not switch to SLIP/PPP; the remote end has no IP address.
24	Exit-Raw-TCP	Disconnect due to exiting raw TCP.
25	Password-Fail	Bad passwords.
26	Raw-TCP-Disabled	Raw TCP disabled.
27	Control-C-Detected	Control-C detected.
28	EXEC-Process-Destroyed	EXEC process destroyed.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **Note** Codes 40, 41, 42, 43, 44, 45, and 46 apply to PPP sessions.
41	Failed-PPP-LCP-Negotiation	PPP LCP negotiation failed.
42	Failed-PPP-PAP-Auth-Fail	PPP PAP authentication failed.
43	Failed-PPP-CHAP-Auth	PPP CHAP authentication failed.
44	Failed-PPP-Remote-Auth	PPP remote authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.
46	PPP-Closed-Event	Upper layer requested that the session be closed.
63	PPP-Echo-Replies	TCP connection has been closed.
100	Session-Timeout	Session timed out.
101	Session-Failed-Security	Session failed for security reasons.
102	Session-End-Callback	Session terminated due to callback.
120	Invalid-Protocol	Call refused because the detected protocol is disabled.
600	VPN-User-Disconnect	Call disconnected by client (through PPP). Code is sent if the LNS receives a PPP terminate request from the client.
601	VPN-Carrier-Loss	Loss of carrier. This can be the result of a physical line going dead. Code is sent when a client is unable to dial out using a dialer.
602	VPN-No-Resources	No resources available to handle the call. Code is sent when the client is unable to allocate memory (running low on memory).

Cause Code	Value	Description
603	VPN-Bad-Control-Packet	Bad L2TP or L2F control packets. This code is sent when an invalid control packet, such as missing mandatory Attribute-Value pairs (AVP), from the peer is received. When using L2TP, the code will be sent after six retransmits; when using L2F, the number of retransmits is user configurable. **Note** VPN-Tunnel-Shut will be sent if there are active sessions in the tunnel.
604	VPN-Admin-Disconnect	Administrative disconnect. This can be the result of a VPN soft shutdown, which is when a client reaches maximum session limit or exceeds maximum hopcount. Code is sent when a tunnel is brought down by issuing the **clear vpdn tunnel** command.
605	VPN-Tunnel-Shut	Tunnel teardown or tunnel setup has failed. Code is sent when there are active sessions in a tunnel and the tunnel goes down. **Note** This code is not sent when tunnel authentication fails.
606	VPN-Local-Disconnect	Call is disconnected by LNS PPP module. Code is sent when the LNS sends a PPP terminate request to the client. It indicates a normal PPP disconnection initiated by the LNS.
607	VPN-Session-Limit	VPN soft shutdown is enabled. Code is sent when a call has been refused due to any of the soft shutdown restrictions previously mentioned.
611	VPDN-Tunnel-In-Resync	VPDN tunnel is in HA resync.

Additional References

The following sections provide references related to RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	*Cisco IOS Security Command Reference*
Security Features	*Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
Security Server Protocols	Security Server Protocols section of the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2

Related Topic	Document Title
RADIUS Configuration	Configuring RADIUS feature module.

Standards

Standard	Title
Internet Engineering Task Force (IETF) Internet Draft: Network Access Servers Requirements	Network Access Servers Requirements: Extended RADIUS Practices

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	Remote Authentication Dial In User Service (RADIUS)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

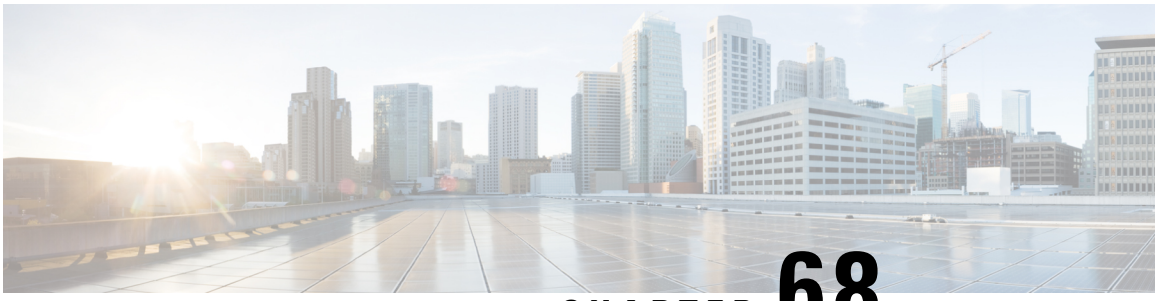
Feature Information for RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 100: Feature Information for RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values

Feature Name	Releases	Feature Information
Accounting of VPDN Disconnect Cause	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Vendor-Specific RADIUS Attributes	Cisco IOS XE Release 2.1	This document discusses the Internet Engineering Task Force (IETF) draft standard, which specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 68

Connect-Info RADIUS Attribute 77

The Connect-Info RADIUS Attribute 77 feature enables the Network Access Server (NAS) to report Connect-Info (attribute 77) in RADIUS accounting "start" and "stop" records that are sent to the RADIUS client (dial-in modem). These records allow the transmit and receive connection speeds, modulation, and compression to be compared in order to analyze a user session over a dial-in modem where speeds are often different at the end of the connection (after negotiation).

When the network access server (NAS) sends attribute 77 in accounting "start" and "stop" records, the connect rates can be measured across the platform. The "transmit" speed (the speed at which the NAS modem sends information) and "receive" speed (the speed at which the NAS receives information) can be recorded to determine whether user modem connections renegotiate to lower speeds shortly into a session. If the transmit and receive speeds are different from each other, attribute 77 reports both speeds, which allows the modem connection speeds that each customer gets from their session.

Attribute 77 is also used to send the Class string for broadband connections such as PPPoX, physical connection speeds for dial access, and the VRF string for any sessions on router interfaces defined with **ip vrf forwarding** command.

Note This feature requires no configuration.

Prerequisites for Connect-Info RADIUS Attribute 77

For information about release and platform support, see the Feature Information for Connect-Info RADIUS Attribute 77, on page 736.

Before the NAS can send attribute 77 in accounting "start" and "stop" records, you must perform the following tasks:

- Configure your NAS for authentication, authorization, and accounting (AAA) and to accept incoming modem calls.

- Enable AAA accounting by using the **aaa accounting network default start-stop group radius** command in global configuration mode.
- Change the modem poll timer by using the **modem link-info poll time** command in global configuration mode.

Note Changing the modem poll timer is required on the Cisco ASR 1000 Series Aggregation Services Routers.

Information About Connect-Info RADIUS Attribute 77

The Configurable Connect-Info Attributes feature introduces support for RADIUS attribute 77 (Connect-Info), which provides information about connection speeds, modulation, and compression for modem dial-in connections via RADIUS accounting “start” and “stop” records.

Customizing Attribute 77 for Ethernet Connections

To customize Attribute 77 for Ethernet connections, enter the connection information as the name of the service policy attached to the Ethernet subinterface. The router takes the policy name and copies it to Attribute 77.

For example, in the following configuration the outbound service policy named speed:eth:25100:5100:19/0 is attached to the QinQ Gigabit Ethernet subinterface 1/0/0.2696. The router copies the policy name to Attribute 77 and sends it to the RADIUS server in an Access-Request or Accounting-Start or Stop message.

```
interface GigabitEthernet1/0/0.2696
encapsulation dot1q 2696 second-dot1q 256
pppoe enable group global
no snmp trap link-status
service-policy input set_precedence_to_0
```

service-policy output speed:eth:25100:5100:19/0

Customizing Attribute 77 for ATM Connections

To customize Attribute 77 for ATM connections, configure the **aaa connect-info** *string* command in the following configuration modes:

- PVC (for a specific PVC)
- PVC range (for a range of PVCs)
- PVC-in-range (for a specific PVC in a range of PVCs)
- VC class (under a specific **class-vc** command)

The router takes the name of the VC class you specify under the **class-vc** command or the string you specify in the **aaa connect-info** *string* command and copies it to Attribute 77.

For example, in the following configuration the **class-vc** command is configured on both ATM PVCs 10/42 and 10/43 and the **aaa connect-info** command is configured on PVC 10/42:

```
interface ATM1/0/0.1 multipoint
description TDSL clients - default TDSL 1024 no ip mroute-cache
class-int speed:ubr:1184:160:10
range pvc 10/41 10/160
!
pvc-in-range 10/42
class-vc speed:ubr:2303:224:10
aaa connect-info speed:ubr:2303:224:10:isp-specific-descr
!
pvc-in-range 10/43
class-vc speed:ubr:2303:224:10
```

For PVC 10/42, the router takes the string (speed:ubr:2303:224:10:isp-specific-descr) specified in the **aaa connect-info** command and copies it to Attribute 77. If the **aaa connect-info** command is not configured on the subinterface, the router takes the class name (speed:ubr:2303:224:10) specified in the **class-vc** command and copies it to Attribute 77.

For PVC 10/43, the router takes the class name (speed:ubr:2303:224:10) specified in the **class-vc** command and copies it to Attribute 77.

How to Verify the Connect-Info RADIUS Attribute 77

Verifying the Connect-Info RADIUS Attribute 77

To verify attribute 77 in your accounting "start" and "stop" records, use the **debug radius** command in privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.

Example

The following example shows the Connect-Info [77] accounting attributes:

```
Router# debug radius
Sep 8 21:53:05.242: RADIUS/ENCODE(00007D34):Orig. component type = PPPoE
Sep 8 21:53:05.242: RADIUS: AAA Unsupported Attr: interface [208] 10
Sep 8 21:53:05.242: RADIUS: 30 2F 31 2F 30 2F 39 2E [ 0/1/0/9.]
Sep 8 21:53:05.242: RADIUS: AAA Unsupported Attr: client-mac-address[45] 14
Sep 8 21:53:05.242: RADIUS: 30 30 30 30 2E 63 30 30 31 2E 30 31 [ 0000.c001.01]
Sep 8 21:53:05.242: RADIUS(00007D34): Config NAS IP: 0.0.0.0
Sep 8 21:53:05.242: RADIUS/ENCODE(00007D34): acct_session_id: 32042
Sep 8 21:53:05.242: RADIUS(00007D34): sending
Sep 8 21:53:05.242: RADIUS/ENCODE: Best Local IP-Address 10.3.8.2 for Radius-Server 10.3.1.107

Sep 8 21:53:05.242: RADIUS(00007D34): Send Access-Request to 10.3.1.107:1645 id 1645/1, len
 116
Sep 8 21:53:05.242: RADIUS: authenticator FC 82 50 DB 65 8F 21 A9 - F3 0A A8 09 29 E5 56
65
Sep 8 21:53:05.242: RADIUS: Framed-Protocol [7] 6 PPP [1]
Sep 8 21:53:05.242: RADIUS: User-Name [1] 8 ''user1''
Sep 8 21:53:05.242: RADIUS: User-Password [2] 18 *
Sep 8 21:53:05.242: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
Sep 8 21:53:05.242: RADIUS: NAS-Port [5] 6 0
Sep 8 21:53:05.242: RADIUS: NAS-Port-Id [87] 12 ''0/1/0/9.32''
Sep 8 21:53:05.242: RADIUS: Connect-Info [77] 28 ''speed:ubr:3456:448:10/0000''
Sep 8 21:53:05.242: RADIUS: Service-Type [6] 6 Framed [2]
Sep 8 21:53:05.242: RADIUS: NAS-IP-Address [4] 6 10.3.8.2
Sep 8 21:53:05.242: RADIUS(00007D34): Started 5 sec timeout
Sep 8 21:53:05.244: RADIUS: Received from id 1645/1 10.3.1.107:1645, Access-Accept, len 32

Sep 8 21:53:05.244: RADIUS: authenticator 9A F1 29 01 66 53 17 CB - 73 FB 1B CE 7D 80 04
F2
Sep 8 21:53:05.244: RADIUS: Service-Type [6] 6 Framed [2]
Sep 8 21:53:05.244: RADIUS: Framed-Protocol [7] 6 PPP [1]
Sep 8 21:53:05.244: RADIUS(00007D34): Received from id 1645/1
Sep 8 21:53:05.248: RADIUS/ENCODE(00007D34):Orig. component type = PPPoE
Sep 8 21:53:05.248: RADIUS(00007D34): Config NAS IP: 0.0.0.0
Sep 8 21:53:05.248: RADIUS(00007D34): sending
Sep 8 21:53:05.248: RADIUS/ENCODE: Best Local IP-Address 10.3.8.2 for Radius-Server 5.3.1.107

Sep 8 21:53:05.248: RADIUS(00007D34): Send Accounting-Request to 10.3.1.107:1646 id 1646/3,
 len 126
Sep 8 21:53:05.248: RADIUS: authenticator 71 6E 73 9B FD 7E 82 81 - 10 2A CD 83 A8 BD D2
F0
Sep 8 21:53:05.248: RADIUS: Acct-Session-Id [44] 10 ''00007D2A''
Sep 8 21:53:05.248: RADIUS: Framed-Protocol [7] 6 PPP [1]
Sep 8 21:53:05.248: RADIUS: User-Name [1] 8 ''user1''
Sep 8 21:53:05.248: RADIUS: Acct-Authentic [45] 6 RADIUS [1]
Sep 8 21:53:05.248: RADIUS: Acct-Status-Type [40] 6 Start [1]
Sep 8 21:53:05.248: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
Sep 8 21:53:05.248: RADIUS: NAS-Port [5] 6 0
Sep 8 21:53:05.248: RADIUS: NAS-Port-Id [87] 12 ''0/1/0/9.32''
Sep 8 21:53:05.248: RADIUS: Connect-Info [77] 28 ''speed:ubr:3456:448:10/0000
```

Configuration Example for Connect-Info RADIUS Attribute 77

Example: Configure NAS for AAA and Incoming Modem Calls

The following example is a sample NAS configuration for AAA and incoming modem calls:

```
interface Serial0:15
  no ip address
  isdn switch-type primary-net5
  isdn incoming-voice modem
!
interface Async1
  ip address 192.0.2.2 255.255.255.0
  encapsulation ppp
  async default routing
  async mode interactive
  no peer default ip address
  ppp authentication chap
!
line 1
  modem InOu
  transport preferred none
  transport input all
  autoselect ppp
!
```

Additional References

The following sections provide references related to the Connect-Info RADIUS Attribute 77 feature.

Related Documents

Related Topic	Document Title
IOS dial technologies	Cisco IOS XE Dial Technologies Configuration Guide, Release 2
	Cisco IOS Dial Technologies Command Reference
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2869	RADIUS Extensions

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

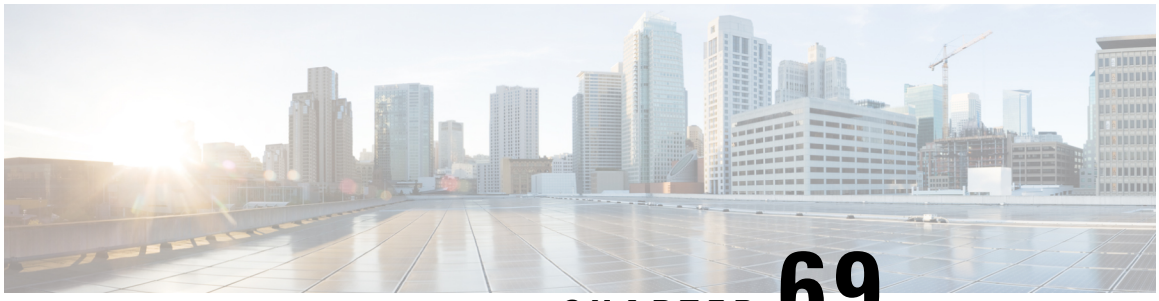
Feature Information for Connect-Info RADIUS Attribute 77

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 101: Feature Information for Connect-Info RADIUS Attribute 77

Feature Name	Releases	Feature Information
Connect-Info RADIUS Attribute 77	Cisco IOS XE Release 2.1	The Connect-Info RADIUS Attribute 77 feature enables the network access server (NAS) to report Connect-Info (attribute 77) in RADIUS accounting “start” and “stop” records that are sent to the RADIUS client (dial-in modem). These “start” and “stop” records allow the transmit and receive connection speeds, modulation, and compression to be compared in order to analyze a user session over a dial-in modem where speeds are often different at the end of the connection (after negotiation). In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 series routers.



CHAPTER 69

Encrypted Vendor-Specific Attributes

The Encrypted Vendor-Specific Attributes feature provides users with a way to centrally manage filters at a RADIUS server and supports the following types of string vendor-specific attributes (VSAs):

- Tagged String VSA, on page 737 (similar to Cisco VSA type 1 (Cisco:AVPair (1)) except that this new VSA is tagged)
- Encrypted String VSA, on page 738 (similar to Cisco VSA type 1 except that this new VSA is encrypted)
- Tagged and Encrypted String VSA, on page 738 (similar to Cisco VSA type 1 except that this new VSA is tagged and encrypted)

Cisco:AVPairs specify additional authentication and authorization information in the form an Attribute-Value Pair (AVPair) string. When Internet Engineering Task Force (IETF) RADIUS attribute 26 (Vendor-Specific) is transmitted with a vendor-Id number of "9" and a vendor-type value of "1" (which means that it is a Cisco AVPair), the RADIUS user profile format for a Cisco AVPair looks as follows: Cisco:AVPair = "protocol:attribute=value".

Prerequisites for Encrypted Vendor-Specific Attributes

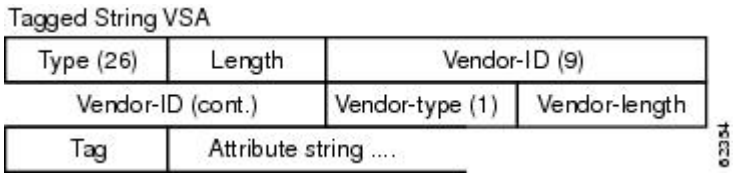
Before the RADIUS server can accept tagged and encrypted VSAs, you must configure your server for AAA authentication and authorization and to accept PPP calls.

Information About Encrypted Vendor-Specific Attributes

Tagged String VSA

The figure below displays the packet format for the Tagged String VSA:

Figure 12: Tagged String VSA Format

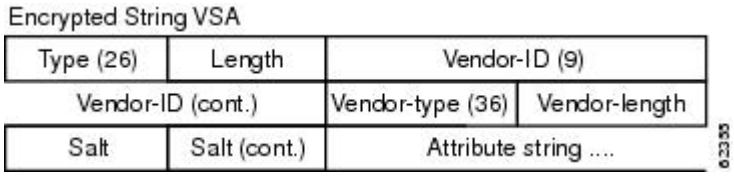


To retrieve the correct value, the Tag field must be parsed correctly. The value for this field can range only from 0x01 through 0x1F. If the value is not within the specified range, the RADIUS server ignores the value and considers the Tag field to be a part of the Attribute String field.

Encrypted String VSA

The figure below displays the packet format for the Encrypted String VSA:

Figure 13: Encrypted String VSA Format



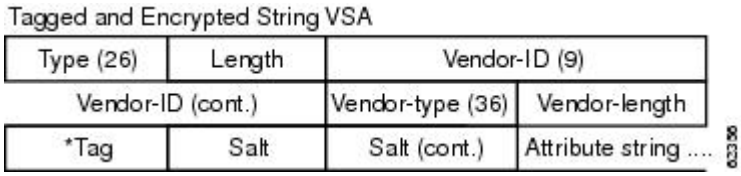
The Salt field ensures the uniqueness of the encryption key that is used to encrypt each instance of the VSA. The first and most significant bit of the Salt field must be set to 1.

Note Vendor-type (36) indicates that the attribute is an encrypted string VSA.

Tagged and Encrypted String VSA

The figure below displays the packet formats for each of the newly supported VSAs:

Figure 14: Tagged and Encrypted String VSA Format



This VSA is similar to encrypted string VSAs except this VSA has an additional Tag field. If the Tag field is not within the valid range (0x01 through 0x1F), it is considered to be part of the Salt field.

How to Verify Encrypted Vendor-Specific Attributes

The Encrypted Vendor-Specific Attributes feature requires no configuration. To verify that RADIUS-tagged and encrypted VSAs are being sent from the RADIUS server, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether tagged and encrypted VSAs are being sent from the RADIUS server.

Configuration Examples for Encrypted Vendor-Specific Attributes

NAS Configuration Example

The following example shows how to configure a network access server (NAS) with a basic configuration using tagged and encrypted VSAs. (This example assumes that the configuration required to make PPP calls is already enabled.)

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
!
radius-server host 10.2.2.2 auth-port 1645 acct-port 1646
radius-server key cisco
```

RADIUS User Profile with a Tagged and Encrypted VSA Example

The following is an example of user profile on a RADIUS server that supports tagged and encrypted string VSAs:

```
mascot   Password = "password1"
         Service-Type = NAS-Prompt,
         Framed-Protocol = PPP,
         Cisco:Cisco-Enc = "ip:route=10.0.0.0 255.0.0.0"
         Cisco.attr Cisco-Enc 36 tag-encstr(*,*)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
RADIUS Attributes	*Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
Media-Independent PPP and Multilink PPP	Configuring Media-Independent PPP and Multilink PPP feature module.
Authentication	Configuring Authentication feature module.
Authorization	Configuring Authorization feature module.

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial In User Service (RADIUS)*
RFC 2868	*RADIUS Attributes for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

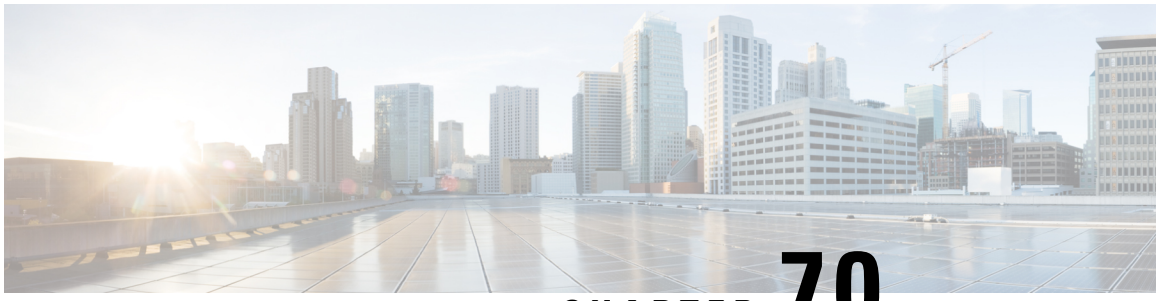
Feature Information for Encrypted Vendor-Specific Attributes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 102: Feature Information for Encrypted Vendor-Specific Attributes

Feature Name	Releases	Feature Information
Encrypted Vendor-Specific Attributes	Cisco IOS XE Release 2.3	The Encrypted Vendor-Specific Attributes feature provides users with a way to centrally manage filters at a RADIUS server and supports the Tagged String, Encrypted String, and Tagged and Encrypted String vendor-specific attributes (VSAs). In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 70

RADIUS Attribute 8 Framed-IP-Address in Access Requests

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible for a network access server (NAS) to provide the RADIUS server with a hint of the user IP address in advance of user authentication. An application can be run on the RADIUS server to use this hint and build a table (map) of user names and addresses. Using the mapping information, service applications can begin preparing user login information to have available upon successful user authentication.

Prerequisites for RADIUS Attribute 8 Framed-IP-Address in Access Requests

Sending RADIUS attribute 8 in the RADIUS access requests assumes that the login host has been configured to request its IP address from the NAS server. It also assumes that the login host has been configured to accept an IP address from the NAS.

The NAS must be configured with a pool of network addresses on the interface supporting the login hosts.

Information About RADIUS Attribute 8 Framed-IP-Address in Access Requests

How This Feature Works

When a network device dials in to a NAS that is configured for RADIUS authentication, the NAS begins the process of contacting the RADIUS server in preparation for user authentication. Typically, the IP address of the dial-in host is not communicated to the RADIUS server until after successful user authentication.

Communicating the device IP address to the server in the RADIUS access request allows other applications to begin to take advantage of that information.

As the NAS is setting up communication with the RADIUS server, the NAS assigns an IP address to the dial-in host from a pool of IP addresses configured at the specific interface. The NAS sends the IP address of the dial-in host to the RADIUS server as attribute 8. At that time, the NAS sends other user information, such as the user name, to the RADIUS server.

After the RADIUS server receives the user information from the NAS, it has two options:

- If the user profile on the RADIUS server already includes attribute 8, the RADIUS server can override the IP address sent by the NAS with the IP address defined as attribute 8 in the user profile. The address defined in the user profile is returned to the NAS.
- If the user profile does not include attribute 8, the RADIUS server can accept attribute 8 from the NAS, and the same address is returned to the NAS.

The address returned by the RADIUS server is saved in memory on the NAS for the life of the session. If the NAS is configured for RADIUS accounting, the accounting start packet sent to the RADIUS server includes the same IP address as in attribute 8. All subsequent accounting packets, updates (if configured), and stop packets will also include the same IP address provided in attribute 8.

However, the RADIUS attribute 8 (Framed-IP-Address) is not included in the accounting start packets in the following two conditions:

- If the user is a dual-stack (IPv4 or IPv6) subscriber.
- If the IP address is from a local pool and not from the RADIUS server.

In both these conditions, use the **aaa accounting delay-start extended-time** *delay-value* command to delay the Internet Protocol Control Protocol version 6 (IPCPv6) address negotiation using the configured delay value. During the delay, the IPCPv4 address is posted and the framed IPv4 address is added to the accounting start packet.

Benefits

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible to run applications on the RADIUS server that builds mapping tables of users and IP addresses. The server can then use the mapping table information in other applications, such as preparing customized user login pages in advance of a successful user authentication with the RADIUS server.

How to Configure RADIUS Attribute 8 Framed-IP-Address in Access Requests

Configuring RADIUS Attribute 8 in Access Requests

To send RADIUS attribute 8 in the access request, perform the following steps:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **radius-server attribute 8 include-in-access-req**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute 8 include-in-access-req** **Example:** `Router(config)# radius-server attribute 8 include-in-access-req`	Sends RADIUS attribute 8 in access-request packets.

Verifying RADIUS Attribute 8 in Access Requests

To verify that RADIUS attribute 8 is being sent in access requests, perform the following steps. Attribute 8 should be present in all PPP access requests.

SUMMARY STEPS

1. **enable**
2. **more system:running-config**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**more system:running-config** **Example:** `Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)

	Command or Action	Purpose
Step 3	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 8 is being sent in access requests.

Configuration Examples for RADIUS Attribute 8 Framed-IP-Address in Access Requests

NAS Configuration That Sends the IP Address of the Dial-in Host Example

The following example shows a NAS configuration that sends the IP address of the dial-in host to the RADIUS server in the RADIUS access request. The NAS is configured for RADIUS authentication, authorization, and accounting (AAA). A pool of IP addresses (async1-pool) has been configured and applied at interface virtual-template1.

```
aaa new-model
aaa authentication login default group radius
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
!
ip address-pool local
!
interface virtual-template1
 peer default ip address pool async1-pool
!
ip local pool async1-pool 209.165.200.225 209.165.200.229
!
radius-server host 172.31.71.146 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 8 include-in-access-req
radius-server key radhost<xxx>: Example
```

Additional References

The following sections provide references related to the RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature.

Related Documents

Related Topic	Document Title
Configuring authentication and c onfiguring RADIUS	“Configuring Authentication” and “Configuring RADIUS” chapters in the *Cisco IOS XE Security Configuration Guide: Configuring User Services* , Release 2.
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2138	Remote Authentication Dial In User Service (RADIUS)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

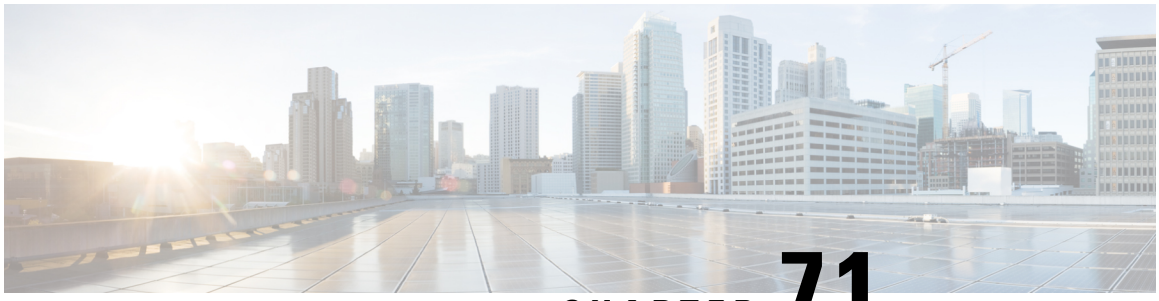
Feature Information for RADIUS Attribute 8 Framed-IP-Address in Access Requests

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 103: Feature Information for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

Feature Name	Releases	Feature Information
RADIUS Attribute 8 (Framed-IP-Address) in Access Requests (Also called Sticky IP)	Cisco IOS XE Release 2.1	The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible for a network access server (NAS) to provide the RADIUS server with a hint of the user IP address in advance of user authentication. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **radius-server attribute 8 include-in-access-req**.



CHAPTER 71

RADIUS Attribute 82 Tunnel Assignment ID

Prerequisites for RADIUS Attribute 82 Tunnel Assignment ID

You must be using a Cisco platform that supports VPDN to use this feature.

Restrictions for Radius Attribute 82 Tunnel Assignment ID

This feature is designed only for VPDN dial-in applications. It does not support VPDN dial-out.

Information about RADIUS Attribute 82 Tunnel Assignment ID

The RADIUS Attribute 82: Tunnel Assignment ID feature allows the Layer 2 Transport Protocol access concentrator (LAC) to group users from different per-user or domain RADIUS profiles into the same active tunnel. The RADIUS Attribute 82: Tunnel Assignment ID feature defines a new avpair, Tunnel-Assignment-ID, which allows the LAC to group users from different RADIUS profiles into the same tunnel if the chosen endpoint, tunnel type, and Tunnel-Assignment-ID are identical. This feature introduces new software functionality. No new commands are introduced with this feature.

How to Verify if RADIUS Attribute 82 is Being Used by the LAC

There are no configuration steps for the RADIUS Attribute 82: Tunnel Assignment ID feature. This task verifies the RADIUS attribute 82 used by the LAC during tunnel authorization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router# **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router# **debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 82 is being sent in access requests.

Configuration Examples for RADIUS Attribute 82 Tunnel Assignment ID

LAC Configuration Example

The following example shows a sample LAC configuration when the VPDN group is defined on the router:

```
aaa new-model
aaa authentication ppp default local
aaa authorization network default local
!
bba-group pppoe bba_group1
virtual-template 1
!
interface Loopback1
no ip address
vpdn-group VPDN_LAC1
request-dialin
protocol l2tp
local name tb162_LAC1
domain isp1.com
initiate-to ip 10.0.0.2
source-ip 10.0.0.1
l2tp tunnel receive-window 100
l2tp tunnel nosession-timeout 30
```

```
l2tp tunnel retransmit retries 5
l2tp tunnel retransmit timeout min 2
l2tp tunnel retransmit timeout max 8
l2tp tunnel hello 60
l2tp tunnel password tunnel1
!
!
interface virtual-template 1
no snmp trap link-status
no keepalive
ip unnumbered loopback1
ppp mtu adaptive
ppp authentication pap
no logging event link-status
!
```

The following example shows a sample LAC configuration when the VPDN group is defined in RADIUS:

```
aaa authentication ppp default group radius
aaa authorization network default radius
!
bba-group pppoe bba_group1
virtual-template 1
!
interface Loopback1
no ip address
interface virtual-template 1
no snmp trap link-status
no keepalive
ip unnumbered loopback1
ppp mtu adaptive
ppp authentication pap
no logging event link-status
```

LNS Configuration Example

The following example configures VPDN on the LNS:

```
hostname lns
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
vpdn enable
vpdn-group VPDN_LNS1
 accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname tb162_LAC1
 local name LNS1
 l2tp tunnel hello 90
 l2tp tunnel password 0 hello1
interface Loopback0
 ip address 10.1.1.3 255.255.255.0
interface Virtual-Template1
 ip unnumbered Loopback0
 no keepalive
 peer default ip address pool mypool
 ppp authentication chap
ip local pool mypool 10.1.1.10 10.1.1.50
radius-server host lns-radiusd  auth-port 1645 acct-port 1646
```

```
radius-server retransmit 3
radius-server key cisco
```

RADIUS Configuration Example

The following examples configure the RADIUS server to group sessions in a tunnel:

Per-User Configuration

```
user@router.com Password = "cisco" Service-Type = Outbound,
       Tunnel-Type = :1:L2TP,
       Tunnel-Server-Endpoint = :1:"10.14.10.54",
       Tunnel-Assignment-Id = :1:"router"
client@router.com Password = "cisco" Service-Type = Outbound,
       Tunnel-Type = :1:L2TP,
       Tunnel-Server-Endpoint = :1:"10.14.10.54",
       Tunnel-Assignment-Id = :1:"router"
```

Domain Configuration

```
eng.router.com Password = "cisco" Service-Type = Outbound,
        Tunnel-Type = :1:L2TP,
        Tunnel-Server-Endpoint = :1:"10.14.10.54",
        Tunnel-Assignment-Id = :1:"router"
sales.router.com Password = "cisco" Service-Type = Outbound,
        Tunnel-Type = :1:L2TP,
        Tunnel-Server-Endpoint = :1:"10.14.10.54",
        Tunnel-Assignment-Id = :1:"router"
```

Additional References

The following sections provide references related to RADIUS Tunnel Attribute Extensions.

Related Documents

Related Topic	Document Title
Authentication	“ Configuring Authentication ” module.
RADIUS Attributes	“ RADIUS Attributes Overview and RADIUS IETF Attributes ” module.
Virtual private dialup networks (VPDN)	*Cisco IOS VPDN Configuration Guide* , Release 15.0.

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2868	RADIUS Attributes for Tunnel Protocol Support

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

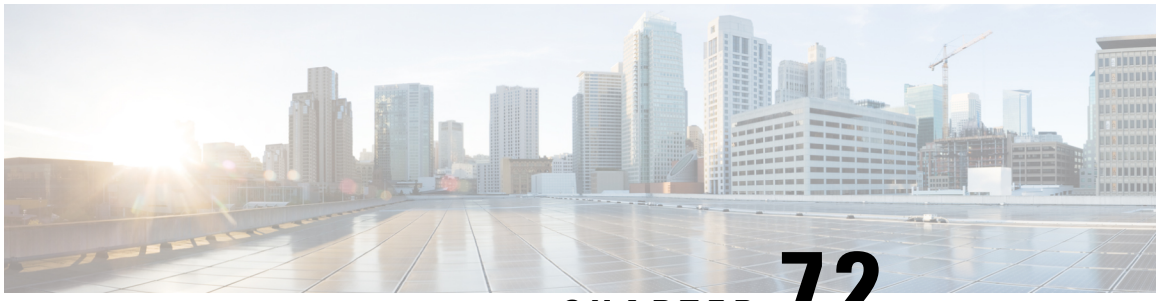
Feature Information for RADIUS Attribute 82 Tunnel Assignment ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 104: Feature Information for RADIUS Attribute 82: Tunnel Assignment ID

Feature Name	Releases	Feature Information
RADIUS Attribute 82: Tunnel Assignment Id	Cisco IOS XE Release 2.1	The RADIUS Attribute 82: Tunnel Assignment ID feature allows the Layer 2 Transport Protocol access concentrator (LAC) to group users from different per-user or domain RADIUS profiles into the same active tunnel. In Cisco IOS XE Release 2.1, support was added for the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 72

RADIUS Tunnel Attribute Extensions

The RADIUS Tunnel Attribute Extensions feature introduces RADIUS attribute 90 (Tunnel-Client-Auth-ID) and RADIUS attribute 91 (Tunnel-Server-Auth-ID). Both attributes help support the provision of compulsory tunneling in virtual private networks (VPNs) by allowing the user to specify authentication names for the network access server (NAS) and the RADIUS server.

Prerequisites

To use RADIUS attributes 90 and 91, you must complete the following tasks:

- Configure your NAS to support AAA.
- Configure your NAS to support RADIUS.
- Configure your NAS to support VPN.

Restrictions

Your RADIUS server must support tagged attributes to use RADIUS tunnel attributes 90 and 91.

Information About RADIUS Tunnel Attribute Extensions

RADIUS Tunnel Attribute Extension Benefits

The RADIUS Tunnel Attribute Extensions feature allows you to specify a name (other than the default) of the tunnel initiator and the tunnel terminator. Thus, you can establish a higher level of security when setting up VPN tunneling.

RADIUS Tunnel Attribute Extension Description

Once a NAS has set up communication with a RADIUS server, you can enable a tunneling protocol. Some applications of tunneling protocols are voluntary, but others involve compulsory tunneling; that is, a tunnel is created without any action from the user and without allowing the user any choice in the matter. In those cases, new RADIUS attributes are needed to carry the tunneling information from the NAS to the RADIUS server to establish authentication. These new RADIUS attributes are listed in the table below.

Note In compulsory tunneling, any security measures in place apply only to traffic between the tunnel endpoints. Encryption or integrity protection of tunneled traffic must not be considered as a replacement for end-to-end security.

Table 105: RADIUS Tunnel Attributes

Number	IETF RADIUS Tunnel Attribute	Equivalent TACACS+ Attribute	Supported Protocols	Description
90	Tunnel-Client-Auth-ID	tunnel-id	Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel initiator (also known as the NAS[5]) when authenticating tunnel setup with the tunnel terminator.
91	Tunnel-Server-Auth-ID	gw-name	Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel terminator (also known as the Home Gateway[6]) when authenticating tunnel setup with the tunnel initiator.

[5] When L2TP is used, the NAS is referred to as an L2TP access concentrator (LAC).
[6] When L2TP is used, the Home Gateway is referred to as an L2TP network server (LNS).

RADIUS attribute 90 and RADIUS attribute 91 are included in the following situations:

- If the RADIUS server accepts the request and the desired authentication name is different from the default, they must be included it.
- If an accounting request contains Acct-Status-Type attributes with values of either start or stop and pertains to a tunneled session, they should be included in.

How to Configure RADIUS Tunnel Attribute Extensions

There are no configuration tasks associated with this feature.

Verifying RADIUS Attribute 90 and RADIUS Attribute 91

To verify that RADIUS attribute 90 and RADIUS attribute 91 are being sent in access accepts and accounting requests, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 90 and attribute 91 are being sent in access accepts and accounting requests.

Configuration Examples for RADIUS Tunnel Attribute Extensions

L2TP Network Server Configuration Example

The following example shows how to configure the LNS with a basic L2F and L2TP configuration using RADIUS tunneling attributes 90 and 91:

```
aaa new-model
aaa authentication login default none
aaa authentication login console none
aaa authentication ppp default local group radius
aaa authorization network default group radius if-authenticated
!
username l2tp-svr-auth-id password 0 l2tp-tnl-pass
!
vpdn enable
vpdn search-order domain
!
vpdn-group 1
accept-dialin
protocol l2tp
virtual-template 1
terminate-from hostname l2tp-cli-auth-id
local name l2tp-svr-auth-id
!
interface loopback0
ip address 10.0.0.3 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface Virtual-Template1
ip unnumbered loopback0
ppp authentication pap
!
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key <deleted>
!
```

RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example

The following is an example of a RADIUS user profile that includes RADIUS tunneling attributes 90 and 91 for an L2TP tunnel.

```
cisco.com Password = "cisco", Service-Type = Outbound
Service-Type = Outbound,
Tunnel-Type = :1:L2TP,
Tunnel-Medium-Type = :1:IP,
Tunnel-Client-Endpoint = :1:"10.0.0.2",
Tunnel-Server-Endpoint = :1:"10.0.0.3",
Tunnel-Client-Auth-Id = :1:"l2tp-cli-auth-id",
Tunnel-Server-Auth-Id = :1:"l2tp-svr-auth-id",
Tunnel-Assignment-Id = :1:"l2tp-assignment-id",
Cisco-Avpair = "vpdn:l2tp-tunnel-password=l2tp-tnl-pass",
Tunnel-Preference = :1:1
```

Additional References

The following sections provide references related to the RADIUS Tunnel Attribute Extensions feature.

Related Documents

Related Topic	Document Title
Authentication configuration	"Configuring Authentication" in the *Cisco IOS XE Security Configuration Guide: Configuring User Services* , Release 2
RADIUS configuration	"Configuring RADIUS" in the *Cisco IOS XE Security Configuration Guide: Configuring User Services* , Release 2
Overview of RADIUS attributes	"RADIUS Attributes Overview and RADIUS IETF Attributes" in the *Cisco IOS XE Security Configuration Guide: Configuring User Services* , Release 2
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2868	*RADIUS Attributes for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Tunnel Attribute Extensions

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 106: Feature Information for RADIUS Tunnel Attribute Extensions

Feature Name	Releases	Feature Information
RADIUS Tunnel Attribute Extensions	Cisco IOS XE Release 2.1	The RADIUS Tunnel Attribute Extensions feature introduces RADIUS attribute 90 (Tunnel-Client-Auth-ID) and RADIUS attribute 91 (Tunnel-Server-Auth-ID). Both attributes help support the provision of compulsory tunneling in virtual private networks (VPNs) by allowing the user to specify authentication names for the network access server (NAS) and the RADIUS server. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Glossary

Layer 2 Tunnel Protocol (L2TP) -- A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

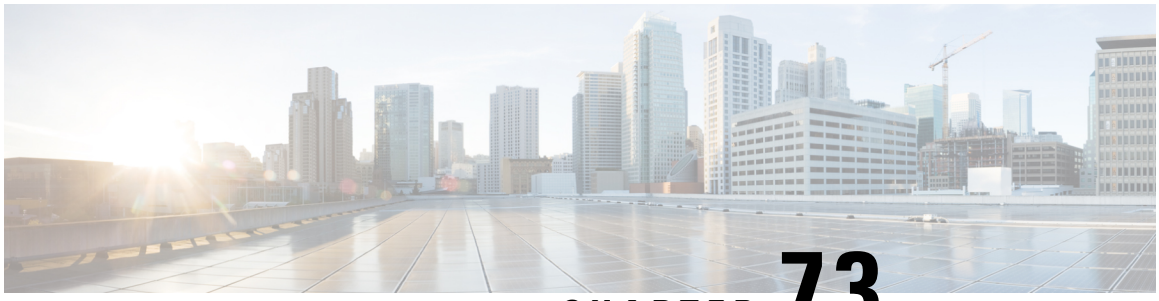
L2TP access concentrator (LAC) --A network access server (NAS) to which the client directly connects and through which PPP frames are tunneled to the L2TP network server (LNS). The LAC need only implement the media over which L2TP is to operate to pass traffic to one or more LNSs. The LAC may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. A LAC is analogous to an L2F network access server.

L2TP network server (LNS) --A termination point for L2TP tunnels, and an access point where PPP frames are processed and passed to higher-layer protocols. An LNS can operate on any platform that terminates PPP. The LNS handles the server side of the L2TP protocol. L2TP relies only on the single medium over which L2TP tunnels arrive. The LNS initiates outgoing calls and receives incoming calls. An LNS is analogous to a home gateway in L2F technology.

network access server (NAS) --A Cisco platform, or collection of platforms, such as an AccessPath system, that interfaces between the packet world (such as the Internet) and the circuit-switched world (such as the PSTN).

tunnel--A virtual pipe between the L2TP access concentrator (LAC) and L2TP network server (LNS) that can carry multiple PPP sessions.

virtual private network (VPN)--A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the L2TP network server (LNS) instead of the L2TP access concentrator (LAC).



CHAPTER 73

RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

The RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements feature allows the hostname of the network access server (NAS) to be specified--rather than the IP address of the NAS--in RADIUS attribute 66 (Tunnel-Client-Endpoint). This feature makes it easier for users to remember a hostname instead of a numerical IP address, and helps disguise the numerical IP address of the NAS.

Prerequisites for RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

A Cisco platform that supports VPDN is required. See the Glossary, on page 764 for more information about VPDN.

Restrictions for RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

Your Cisco device must be running a Cisco software image that supports virtual private dialup networks (VPDNs).

Information About RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

How the RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements are Used

Virtual Private Networks (VPNs) use Layer 2 Forwarding (L2F) or Layer 2 Tunnel Protocol (L2TP) tunnels to tunnel the link layer of high-level protocols (for example, PPP or asynchronous High-Level Data Link Control (HDLC)). Internet service providers (ISPs) configure their NASs to receive calls from users and forward the calls to the customer tunnel server. Usually, the ISP maintains only information about the tunnel server--the tunnel endpoint. The customer maintains the IP addresses, routing, and other user database functions of the tunnel server users. RADIUS attribute 66 provides the customer with the ability to specify the hostname of the NAS instead of the IP address of the NAS.

Note L2F is not supported on the Cisco ASR 1000 Series Aggregation Services Routers.

How to Configure RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

There are no configuration tasks associated with support for the RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements.

Configuration Examples for RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

Setting Up the RADIUS Profile for RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements Example

The following example shows a configuration that allows the user to specify the hostname of the NAS using RADIUS attribute 66 (Tunnel-Client-Endpoint) in the RADIUS profile:

```
cisco-avpair = vpdn:l2tp-cm-local-window-size=1024
cisco-avpair = vpdn:l2tp-nosession-timeout=30
cisco-avpair = vpdn:l2tp-cm-retransmit-retries=10
cisco-avpair = vpdn:l2tp-cm-min-timeout=2
cisco-avpair = vpdn:l2tp-hello-interval=60
Service-Type = outbound
Tunnel-Assignment-Id_tag1 = ISP1
Tunnel-Client-Auth-Id_tag1 = LAC1
Tunnel-Client-Endpoint_tag1 = 10.0.0.2
Tunnel-Medium-Type_tag1 = IPv4
```

```
Tunnel-Password_tag1 = tunnel1
Tunnel-Server-Auth-Id_tag1 = LNS1
Tunnel-Server-Endpoint_tag1 = 10.0.0.1
Tunnel-Type_tag1 = l2tp
```

Additional References

The following sections provide references related to the RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements feature.

Related Documents

Related Topic	Document Title
RADIUS attribute 66	*Cisco IOS XE Security Configuration Guide: Configuring User Services* , Release 2
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Attribute 66 Tunnel-Client-Endpoint Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 107: Feature Information for RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements

Feature Name	Releases	Feature Information
RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements	Cisco IOS XE Release 2.1 Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	The RADIUS Attribute 66 (Tunnel-Client-Endpoint) Enhancements feature allows the hostname of the network access server (NAS) to be specified—rather than the IP address of the NAS—in RADIUS attribute 66 (Tunnel-Client-Endpoint). This feature makes it easier for users to remember a hostname instead of a numerical IP address, and helps disguise the numerical IP address of the NAS. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Glossary

L2F--Layer 2 Forwarding Protocol. Protocol that supports the creation of secure virtual private dialup networks over the Internet.

L2TP--Layer 2 Tunnel Protocol. Protocol that is one of the key building blocks for virtual private networks in the dial access space and is endorsed by Cisco and other internetworking industry leaders. This protocol

combines the best of Cisco's Layer 2 Forwarding (L2F) protocol and Microsoft's Point-to-Point Tunneling Protocol (PPTP).

Layer 2 Forwarding Protocol--See L2F.

Layer 2 Tunnel Protocol--See L2TP.

Point-to-Point Protocol--See PPP.

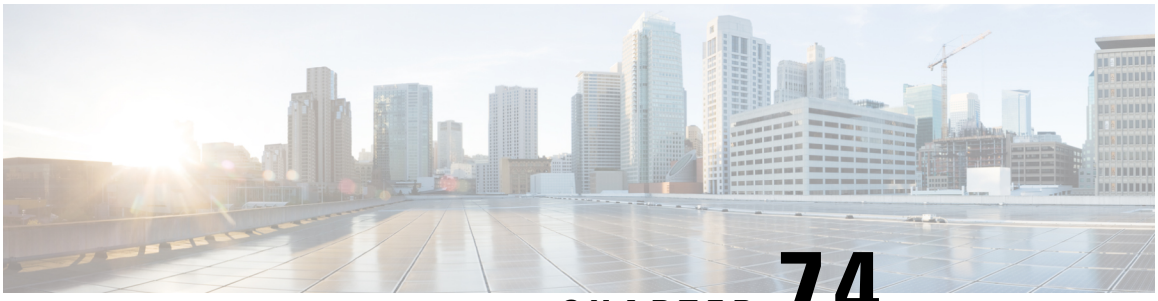
PPP--Point-to-Point Protocol. Successor to SLIP that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. Whereas SLIP was designed to work with IP, PPP was designed to work with several network layer protocols, such as IP, IPX, and ARA. PPP also has built-in security mechanisms, such as CHAP and PAP. PPP relies on two protocols: LCP and NCP.

RADIUS--Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

Remote Authentication Dial-In User Service--See RADIUS.

virtual private dialup network--See VPDN.

VPDN--virtual private dialup network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPDNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the L2TP network server (LNS), instead of the L2TP access concentrator (LAC).



CHAPTER 74

RADIUS Attribute Value Screening

The RADIUS Attribute Value Screening feature allows users to configure a list of "accept" or "reject" RADIUS attributes on the network access server (NAS) for purposes such as authorization or accounting.

If a NAS accepts and processes all RADIUS attributes received in an Access-Accept packet, unwanted attributes may be processed, creating a problem for wholesale providers who do not control their customers' authentication, authorization, and accounting (AAA) servers. For example, there may be attributes that specify services to which the customer has not subscribed, or there may be attributes that may degrade service for other wholesale dial users. The ability to configure the NAS to restrict the use of specific attributes has therefore become a requirement for many users.

The RADIUS Attribute Value Screening feature should be implemented in one of the following ways:

- To allow the NAS to accept and process all standard RADIUS attributes for a particular purpose, except for those on a configured reject list
- To allow the NAS to reject (filter out) all standard RADIUS attributes for a particular purpose, except for those on a configured accept list

Prerequisites for RADIUS Attribute Value Screening

Before configuring a RADIUS accept or reject list, you must enable AAA.

Restrictions for RADIUS Attribute Value Screening

NAS Requirements

To enable this feature, your NAS should be configured for authorization with RADIUS groups.

Accept or Reject Lists Limitations

The two filters used to configure accept or reject lists are mutually exclusive; therefore, a user can configure only one access list or one reject list for each purpose, per server group.

Vendor-Specific Attributes

This feature does not support vendor-specific attribute (VSA) screening; however, a user can specify attribute 26 (Vendor-Specific) in an accept or reject list, which accepts or reject all VSAs.

Required Attributes Screening Recommendation

It is recommended that users do not reject the following required attributes:

- For authorization:
  - 6 (Service-Type)
  - 7 (Framed-Protocol)
- For accounting:
  - 4 (NAS-IP-Address)
  - 40 (Acct-Status-Type)
  - 41 (Acct-Delay-Time)
  - 44 (Acct-Session-ID)

If an attribute is required, the rejection is refused, and the attribute is allowed to pass through.

Note The user does not receive an error at the point of configuring a reject list for required attributes because the list does not specify a purpose--authorization or accounting. The server determines whether an attribute is required when it is known what the attribute is to be used for.

Information About RADIUS Attribute Value Screening

The RADIUS Attribute Value Screening feature provides the following benefits:

- Users can configure an accept or reject list consisting of a selection of attributes on the NAS for a specific purpose so unwanted attributes are not accepted and processed.
- Users may wish to configure an accept list that includes only relevant accounting attributes, thereby reducing unnecessary traffic and allowing users to customize their accounting data.

How to Screen RADIUS Attributes

Configuring RADIUS Attribute Value Screening

To configure a RADIUS attribute accept or reject list for authorization or accounting, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **aaa authentication ppp default**
4. Router(config)# **aaa authorization network default group** *group-name*
5. Router(config)# **aaa group server radius** *group-name*
6. Router(config-sg-radius)# **server** *ip-address*
7. Router(config-sg-radius)# **authorization** [**accept** | **reject**] *listname*
8. Router(config-sg-radius)# **exit**
9. Router(config)# **radius-server host** {*hostname* | *ip-address*} [**key** *string*
10. Router(config)# **radius-server attribute list** *listname*
11. Router(config-sg-radius)# **attribute** *value1* [*value2* [*value3...*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **aaa authentication ppp default** **Example:** **group** *group-name*	Specifies one or more AAA authentication methods for use on serial interfaces running PPP.
Step 4	Router(config)# **aaa authorization network default group** *group-name*	Sets parameters that restrict network access to the user.
Step 5	Router(config)# **aaa group server radius** *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods.
Step 6	Router(config-sg-radius)# **server** *ip-address*	Configures the IP address of the RADIUS server for the group server,
Step 7	Router(config-sg-radius)# **authorization** [**accept** | **reject**] *listname* **Example:** `and/or` **Example:**	Specifies a filter for the attributes that are returned in an Access-Accept packet from the RADIUS server. and/or Specifies a filter for the attributes that are to be sent to the RADIUS server in an accounting request.

	Command or Action	Purpose
	`Router(config-sg-radius)# accounting [accept | reject] listname`	**Note** The **accept** keyword indicates that all attributes are rejected except for the attributes specified in the *listname*. The **reject** keyword indicates that all attributes are accepted except for the attributes specified in the *listname* and all standard attributes.
Step 8	Router(config-sg-radius)# **exit**	Exits server-group configuration mode.
Step 9	Router(config)# **radius-server host** {*hostname* | *ip-address*} [**key** *string*	Specifies a RADIUS server host.
Step 10	Router(config)# **radius-server attribute list** *listname*	Defines the list name given to the set of attributes defined in the **attribute** command. **Note** The *listname* must be the same as the *listname* defined in Step 5.
Step 11	Router(config-sg-radius)# **attribute** *value1* [*value2* [*value3...*]]	Adds attributes to the configured accept or reject list. **Note** This command can be used multiple times to add attributes to an accept or reject list.

Verifying RADIUS Attribute Value Screening

To verify an accept or reject list, use one of the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authentication`	Displays information on AAA authentication.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

Configuration Examples for RADIUS Attribute Value Screening

Authorization Accept Example

The following example shows how to configure an accept list for attribute 6 (Service-Type) and attribute 7 (Framed-Protocol); all other attributes (including VSAs) are rejected for RADIUS authorization.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
```

```
aaa group server radius radius-sg
server 10.1.1.1
authorization accept min-author
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list min-author
attribute 6-7
```

Accounting Reject Example

The following example shows how to configure a reject list for attribute 66 (Tunnel-Client-Endpoint) and attribute 67 (Tunnel-Server-Endpoint); all other attributes (including VSAs) are accepted for RADIUS accounting.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
server 10.1.1.1
accounting reject tnl-x-endpoint
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list tnl-x-endpoint
attribute 66-67
```

Authorization Reject and Accounting Accept Example

The following example shows how to configure a reject list for RADIUS authorization and configure an accept list for RADIUS accounting. Although you cannot configure more than one accept or reject list per server group for authorization or accounting, you can configure one list for authorization and one list for accounting per server group.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
server 10.1.1.1
authorization reject bad-author
accounting accept usage-only
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list usage-only
attribute 1,40,42-43,46
!
radius-server attribute list bad-author
attribute 22,27-28,56-59
```

Rejecting Required Attributes Example

The following example shows debug output for the **debug aaa accounting** command. In this example, required attributes 44, 40, and 41 have been added to the reject list “standard.”

```
Router# debug aaa authorization
AAA/ACCT(6): Accounting method=radius-sg (radius)
RADIUS: attribute 44 cannot be rejected
RADIUS: attribute 61 rejected
```

```
RADIUS: attribute 31 rejected
RADIUS: attribute 40 cannot be rejected
RADIUS: attribute 41 cannot be rejected
```

Additional References

The following sections provide references related to the RADIUS Attribute Value Screening feature.

Related Documents

Related Topic	Document Title
RADIUS	" Configuring RADIUS " feature module.
Other security features	*Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

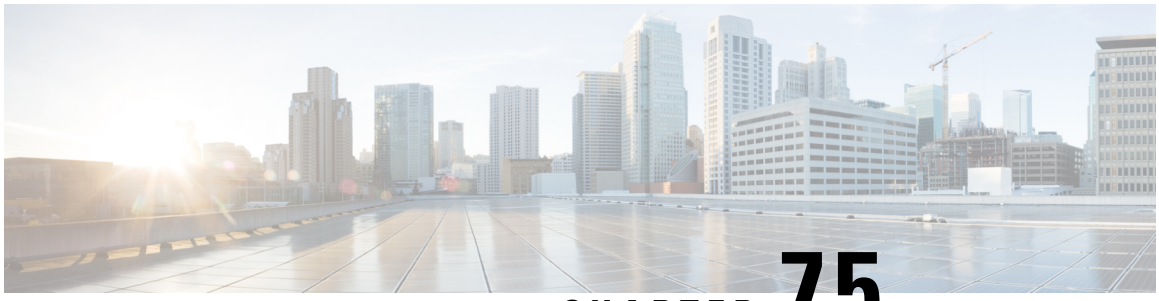
Feature Information for RADIUS Attribute Value Screening

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 108: Feature Information for RADIUS Attribute Value Screening

Feature Name	Releases	Feature Information
RADIUS Attribute Value Screening	Cisco IOS XE Release 2.1	The RADIUS Attribute Value Screening feature allows users to configure a list of “accept” or “reject” RADIUS attributes on the network access server (NAS) for purposes such as authorization or accounting. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers The following commands were introduced or modified by this feature: **accounting (server-group), authorization (server-group), attribute (server-group), radius-server attribute list**



CHAPTER 75

RADIUS Attribute 55 Event-Timestamp

The RADIUS Attribute 55 Event-Timestamp feature allows a network access server (NAS) to insert an event time-stamp attribute in accounting and authentication packets that are sent to the RADIUS server with or without Network Time Protocol (NTP) synchronization.

Prerequisites for RADIUS Attribute 55 Event-Timestamp

Before the Event-Timestamp attribute can be sent in accounting and authentication request packets, you must configure the clock on the network device. For information about setting the clock on your network device, see the "Performing Basic System Management" section in the "Basic System Management" chapter of *Network Management Configuration Guide*.

To avoid configuring the clock on the network device every time the network device is reloaded, you can enable the **clock calendar-valid** command. For information about this command, see the "Setting Time and Calendar Services" section in the "Basic System Management" chapter of *Network Management Configuration Guide*.

Information About RADIUS Attribute 55 Event-Timestamp

When a network device dials in to a network access server (NAS) that is configured for RADIUS authentication, the NAS begins the process of contacting the RADIUS server in preparation for user authentication. Typically, the RADIUS attribute 55 (Event-Timestamp) is not communicated to the RADIUS server until after a successful Network Time Protocol (NTP) synchronization. This feature enables a NAS to insert the Event-Timestamp attribute in accounting and authentication request packets even if NTP synchronization does not happen.

The Event-Timestamp attribute records the time at which the event occurred on the NAS. This times tamp is sent in seconds in RADIUS attribute 55 since January 1, 1970 00:00 UTC.

The Event-Timestamp attribute is saved in memory on the NAS for the life of the session. The RADIUS accounting and authentication start packet, all subsequent accounting and authentication packets, updates (if

configured), and stop packets also include the same RADIUS attribute 55 Event-Timestamp representing the time at which the original packet was sent.

How to Configure RADIUS Attribute 55 Event-Timestamp

Configuring RADIUS Attribute 55 Event-Timestamp

Perform this task to send RADIUS attribute 55 in accounting and authentication requests.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication ppp default group radius**
5. **aaa accounting network default start-stop group radius**
6. **radius-server host** *ip-address*
7. **radius-server attribute 55 include-in-acct-req**
8. **radius-server attribute 55 access-req include**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA).
Step 4	**aaa authentication ppp default group radius** **Example:** `Device(config)# aaa authentication ppp default group radius`	Specifies one or more AAA methods for use on serial interfaces that run PPP using the list of all RADIUS servers for authentication.

	Command or Action	Purpose
Step 5	**aaa accounting network default start-stop group radius** **Example:** `Device(config)# aaa accounting network default start-stop group radius`	Enables network accounting and sends start and stop accounting notices for the RADIUS accounting method list to the RADIUS server.
Step 6	**radius-server host** *ip-address* **Example:** `Device(config)# radius-server host 192.0.2.3`	Specifies the IP address of the RADIUS server host.
Step 7	**radius-server attribute 55 include-in-acct-req** **Example:** `Device(config)# radius-server attribute 55 include-in-acct-req`	Sends RADIUS attribute 55 in account-request packets.
Step 8	**radius-server attribute 55 access-req include** **Example:** `Device(config)# radius-server attribute 55 access-req include`	Sends RADIUS attribute 55 in access-request packets.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Verifying RADIUS Attribute 55 Event-Timestamp

Perform this task to verify that RADIUS attribute 55 is sent in accounting and authentication packets.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **debug radius**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show running-config**

Displays the contents of the current running configuration file.

Example:

```
Device# show running-config

.
.
.
aaa group server radius sample
aaa accounting network default start-stop group radius group sample
aaa server radius dynamic-author
radius-server attribute 55 include-in-acct-req
radius-server attribute 55 access-request include
radius-server dead-criteria time 10 tries 3
radius-server host 192.0.2.3
radius-server retry method reorder
radius-server retransmit 2
radius-server deadtime 1
radius-server key rad123
radius server host
.
.
.
radius-server attribute 55 include-in-acct-req
radius-server attribute 55 access-request include
```

Step 3 **debug radius**

Displays information associated with RADIUS. The output of this command shows whether attribute 55 is being sent in accounting and authentication requests.

Example:

```
Device# debug radius

AAA/BIND(0000000D): Bind i/f Virtual-Template1
AAA/AUTHEN/PPP (0000000D): Pick method list 'default'
RADIUS/ENCODE(0000000D):Orig. component type = PPPoE
RADIUS: DSL line rate attributes successfully added
RADIUS(0000000D): Config NAS IP: 0.0.0.0
RADIUS(0000000D): Config NAS IPv6: ::
RADIUS/ENCODE(0000000D): acct_session_id: 2
RADIUS(0000000D): sending
RADIUS/ENCODE: Best Local IP-Address 192.0.2.3 for Radius-Server 192.0.2.1
RADIUS(0000000D): Sending a IPv4 Radius Packet
RADIUS(0000000D): Send Access-Request to 192.0.2.1:1645 id 1645/1,len 130
RADIUS:  authenticator 66 D8 24 42 BC 45 5B 3D - 0E DC 74 D7 E9 3D 81 85
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS:  User-Name           [1]   6   "test"
RADIUS:  User-Password       [2]   18  *
RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
RADIUS:  NAS-Port            [5]   6   0
RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
RADIUS:  Vendor, Cisco       [26]  41
RADIUS:   Cisco AVpair       [1]   35  "client-mac-address=aabb.cc00.6500"
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  NAS-IP-Address      [4]   6   1.1.1.2
```

```
RADIUS:  Event-Timestamp     [55]  6   1362041578
RADIUS(0000000D): Started 5 sec timeout
RADIUS: Received from id 1645/192.0.2.1:1645, Access-Accept, len 20
.
.
.
RADIUS:  authenticator 2A 2B 24 47 06 44 23 8A - CB CC 8C 96 8D 21 76 DD
RADIUS(0000000D): Received from id 1645/1
AAA/BIND(0000000D): Bind i/f Virtual-Access2.1
RADIUS/ENCODE(0000000D):Orig. component type = PPPoE
.
.
.
RADIUS(0000000D): Config NAS IP: 0.0.0.0
RADIUS(0000000D): Config NAS IPv6: ::
RADIUS(0000000D): sending
RADIUS/ENCODE: Best Local IP-Address 192.0.2.3 for Radius-Server 192.0.2.1
RADIUS(0000000D): Sending a IPv4 Radius Packet
RADIUS(0000000D): Send Accounting-Request to 192.0.2.1:1646 id 1646/1,len 182
RADIUS:  authenticator C6 81 D0 D7 EA BA 9A A9 - 19 4B 1B 90 B8 D1 66 BF
RADIUS:  Acct-Session-Id     [44]  10  "00000002"
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS:  User-Name           [1]   6   "test"
RADIUS:  Vendor, Cisco       [26]  32
RADIUS:   Cisco AVpair       [1]   26  "connect-progress=Call Up"
RADIUS:  Acct-Authentic      [45]  6   RADIUS                    [1]
RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
RADIUS:  NAS-Port            [5]   6   0
RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
RADIUS:  Vendor, Cisco       [26]  41
RADIUS:   Cisco AVpair       [1]   35  "client-mac-address=aabb.cc00.6500"
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  NAS-IP-Address      [4]   6   1.1.1.2
RADIUS:  home-hl-prefix      [151] 10  "163BD6D4"
RADIUS:  Event-Timestamp     [55]  6   1362041588
RADIUS:  Acct-Delay-Time     [41]  6   0
RADIUS(0000000D): Started 5 sec timeout
.
.
.
RADIUS: Received from id 1646/1 1.1.1.1:1646, Accounting-response, len 20
RADIUS:  authenticator 79 F1 6A 38 07 C3 C8 F9 - 96 66 BE EF 5C FA 91 E6
```

Configuration Example for RADIUS Attribute 55 Event-Timestamp

Example: RADIUS Attribute 55 in Accounting and Authentication Packets

The following example shows a configuration that sends RADIUS attribute 55 in accounting and authentication packets:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp default group radius
```

```
Device(config)# aaa accounting network default start-stop group radius
Device(config)# radius-server host 192.0.2.3
Device(config)# radius-server attribute 55 include-in-acct-req
Device(config)# radius-server attribute 55 access-req include
Device(config)# exit
```

Additional References for RADIUS Attribute 55 Event-Timestamp

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Configuring Authentication	"Configuring Authentication" chapter in *Authentication, Authorization, and Accounting Configuration Guide*
Configuring RADIUS	"Configuring RADIUS" chapter in *RADIUS Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2138	*Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

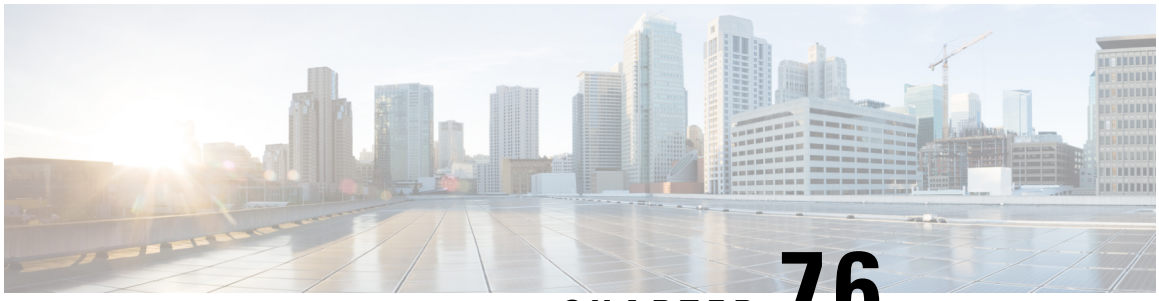
Feature Information for RADIUS Attribute 55 Event-Timestamp

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 109: Feature Information for RADIUS Attribute 55 Event-Timestamp

Feature Name	Releases	Feature Information
RADIUS Attribute 55 Event-Timestamp	Cisco IOS XE Release 3.9S	The RADIUS Attribute 55 Event-Timestamp feature allows a network access server (NAS) to insert an event time-stamp attribute in accounting and authentication packets sent to the RADIUS server with or without Network Time Protocol (NTP) synchronization. The following commands were introduced or modified: **radius-server attribute 55 access-req include** and **radius-server attribute 55 include-in-acct-req**.



CHAPTER 76

RADIUS Attribute 104

The RADIUS Attribute 104 feature allows private routes (attribute 104) to be specified in a RADIUS authorization profile. The private routes affect only packets that are received on an individual interface. The routes are stored apart from the global routing table and are not injected into any routing protocols for redistribution.

Prerequisites for RADIUS Attribute 104

- You must be using a Cisco RADIUS server.
- You should be familiar with configuring RADIUS.
- You should be familiar with policy-based routing (PBR) and private routes.
- You should be familiar with configuring access control lists (ACLs).
- Before using the RADIUS Attribute 104 feature, you must configure RADIUS AAA authorization and RADIUS route download.
- The following memory bytes are requireF:\tips-migration
  - One route map--50 bytes.
  - One match-set clause--600 bytes.
  - One extended ACL--366 bytes.
  - For N number of attribute 104s, the memory requirement is (600+366)*N+50=1000*N(approximate) per user.

Restrictions for RADIUS Attribute 104

- If you already have PBR locally (statically) configured under the interface, and you specify attribute 104, the locally configured PBR will be disabled.
- If a pseudo next-hop address is involved, there must be a route available in the routing table for the next-hop address. If a route is not available, the packet will not be policy routed.
- Policy routing does not order the match-set clauses and relies on the first match, so you should specify the attributes in the order in which you want them to be matched.
- Metric numbers cannot be used in the attribute.

Information About RADIUS Attribute 104

Policy-Based Routing Background

PBR provides a mechanism for the forwarding, or routing of, data packets on the basis of defined policies. The policies are not wholly dependent on the destination address but rather on other factors, such as type of service, source address, precedence, port numbers, or protocol type.

Policy-based routing is applied to incoming packets. All packets that are received on an interface that has policy-based routing enabled are considered for policy-based routing. The router passes the packets through enhanced packet filters called route maps. On the basis of the criteria that are defined in the route maps, the packets are forwarded to the appropriate next hop.

Each entry in a route map statement contains a combination of match clauses and set clauses or commands. The match clauses define the criteria for whether appropriate packets meet the particular policy (that is, whether the conditions are met). The set clauses provide instruction for how the packets should be routed after they have met the match criteria. The match clause specifies which set of filters a packet must match for the corresponding set clause to be applied.

Attribute 104 and the Policy-Based Route Map

This section discusses the attribute 104 feature and how it works with policy-based route maps.

RADIUS Attribute 104 Overview

Using the RADIUS Attribute 104 feature, you can specify private routes in your RADIUS authorization profile. The private routes you specify will affect only packets that are received on an individual interface. The routes are stored apart from the global routing table and are not injected into any routing protocols for redistribution.

Permit Route Map

Route map statements can be marked as “permit” or “deny.” If the statement is marked “permit,” the set clause is applied to the packets that match the match criteria. For attribute 104, when you are configuring the route

map, you need to mark the route map as “permit,” as follows. See Related Documents, on page 788for where to find information on configuring a route map.

Default Private Route

The policy routing process proceeds through the route map until a match is found. If no match is found in the route map, the global routing table is consulted. If you have specified a default route in your user profile, any further routes beyond the default route are effectively ignored.

Route Map Order

You need to specify route maps on the server in the order that you want them to be applied.

How to Apply RADIUS Attribute 104

Applying RADIUS Attribute 104 to Your User Profile

You can apply RADIUS attribute 104 to your user profile by adding the following to the RADIUS server database.

SUMMARY STEPS

1. Apply RADIUS attribute 104 to your user profile.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Apply RADIUS attribute 104 to your user profile.	`Ascend-Private-Route=”dest_addr/netmask next_hop”` The destination network address of the router is “dest_addr/netmask”, and the address of the next-hop router is “next_hop.”

Examples

The following is a sample user profile that creates three private routes that are associated with the caller:

```
username Password="ascend"; User-Service=Framed-User
   Framed-Protocol=PPP,
   Framed-Address=10.1.1.1,
   Framed-Netmask=255.0.0.0,
   Ascend-Private-Route="172.16.1.1/16 10.10.10.1"
   Ascend-Private-Route="192.168.1.1/32 10.10.10.2"
   Ascend-Private-Route="10.20.0.0/1 10.10.10.3"
   Ascend-Private-Route="10.0.0.0/0 10.10.10.4"
```

Using the above profile, the private routing table for the connection contains the following routes, including a default route:

```
Destination/Mask     Gateway
172.16.1.1/16         10.10.10.1
192.168.1.1/32         10.10.10.2
10.20.20.20/1       10.10.10.3
10.0.0.0/0            10.10.10.4
```

Verifying Route Maps

You can use the following **show** commands to verify the route maps that have been configured.

SUMMARY STEPS

1. **enable**
2. **show ip policy**
3. **show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip policy** **Example:** `Router# show ip policy`	Displays the route map that is used for policy routing.
Step 3	**show route-map** [*map-name*	**dynamic** [*dynamic-map-name*	**application** [*application-name*]]	**all**] **Example:** `Router# show route-map`	Displays all route maps that are configured or only the one that is specified.

Troubleshooting the RADIUS Profile

If your private route configuration is not working properly, you may want to reread the section " Policy-Based Routing Background, on page 784." This section may help you determine what is happening to the packets. In addition, the following **debug** commands can be used to troubleshoot your RADIUS profile.

SUMMARY STEPS

1. **enable**
2. **debug radius**
3. **debug aaa per-user**

4. debug ip policy

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.
Step 3	**debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays the attributes that are applied to each user as the user authenticates.
Step 4	debug ip policy **Example:** `Router# debug ip policy`	Displays IP routing packet activity.

Configuration Examples for RADIUS Attribute 104

Route-Map Configuration in Which Attribute 104 Has Been Applied Example

The following output is a typical route-map configuration to which attribute 104 has been applieF:\tips-migration

```
Router# show route-map dynamic
route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 0, identifier 1639994476
  Match clauses:
    ip address (access-lists): PBR#1 PBR#2
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 1, identifier 1640264784
  Match clauses:
    ip address (access-lists): PBR#3 PBR#4
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 2, identifier 1645563704
  Match clauses:
    ip address (access-lists): PBR#5 PBR#6
    length 10 100
  Set clauses:
    ip next-hop 10.1.1.1
    ip gateway10.1.1.1
```

```
 Policy routing matches: 0 packets, 0 bytes
Current active dynamic routemaps = 1
```

Additional References

The following sections provide references related to RADIUS NAS-IP-Address Attribute Configurability.

Related Documents

Related Topic	Document Title
Configuring AAA	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide: Securing User Services*
Configuring RADIUS	" Configuring RADIUS " module.
RADIUS commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

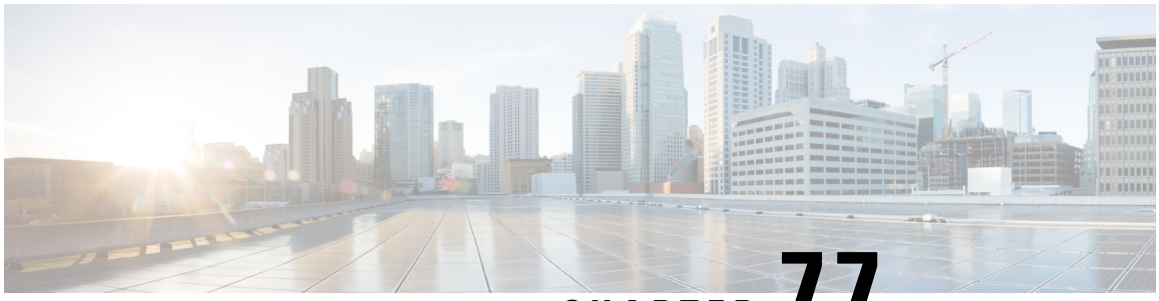
Feature Information for RADIUS Attribute 104

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 110: Feature Information for RADIUS Attribute 104

Feature Name	Releases	Feature Information
RADIUS Attribute 104	Cisco IOS XE Release 3.9S	The RADIUS Attribute 104 feature allows private routes (attribute 104) to be specified in a RADIUS authorization profile. The private routes affect only packets that are received on an individual interface. The routes are stored apart from the global routing table and are not injected into any routing protocols for redistribution. The following commands were introduced or modifieF:\tips-migration **show ip policy**, **show route-map**.



CHAPTER 77

RADIUS NAS-IP-Address Attribute Configurability

The RADIUS NAS-IP-Address Attribute Configurability feature allows an arbitrary IP address to be configured and used as RADIUS attribute 4, NAS-IP-Address, without changing the source IP address in the IP header of the RADIUS packets. This feature may be used for situations in which service providers are using a cluster of small network access servers (NASs) to simulate a large NAS to improve scalability. This feature allows the NASs to behave as a single RADIUS client from the perspective of the RADIUS server.

Prerequisites for RADIUS NAS-IP-Address Attribute Configurability

The following requirements are necessary before configuring this feature:

- Experience with IP Security (IPSec) and configuring both RADIUS servers and authentication, authorization, and accounting (AAA) is necessary.
- RADIUS server and AAA lists must be configured.

Restrictions for RADIUS NAS-IP-Address Attribute Configurability

The following restrictions apply if a cluster of RADIUS clients are being used to simulate a single RADIUS client for scalability. Solutions, or workarounds, to the restrictions are also provided.

- RADIUS attribute 44, Acct-Session-Id, may overlap among sessions from different NASs.

There are two solutions. Either the **radius-server attribute 44 extend-with-addr** or **radius-server unique-ident** command can be used on NAS routers to specify different prepending numbers for different NAS routers.

- RADIUS server-based IP address pool for different NASs must be managed.

The solution is to configure different IP address pool profiles for different NASs on the RADIUS server. Different NASs use different pool usernames to retrieve them.

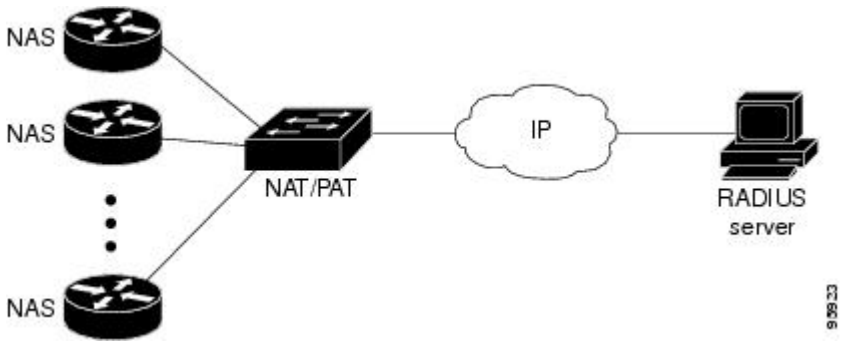
- RADIUS request message for sessions from different NASs must be differentiated.

One of the solutions is to configure different format strings for RADIUS attribute 32, NAS-Identifier, using the **radius-server attribute 32 include-in-access-req** command on different NASs.

Information About RADIUS NAS-IP-Address Attribute Configurability

To simulate a large NAS RADIUS client using a cluster of small NAS RADIUS clients, as shown in the figure below, a Network Address Translation (NAT) or Port Address Translation (PAT) device is inserted in a network. The device is placed between a cluster of NASs and the IP cloud that is connected to a RADIUS server. When RADIUS traffic from different NASs goes through the NAT or PAT device, the source IP addresses of the RADIUS packets are translated to a single IP address, most likely an IP address on a loopback interface on the NAT or PAT device. Different User Datagram Protocol (UDP) source ports are assigned to RADIUS packets from different NASs. When the RADIUS reply comes back from the server, the NAT or PAT device receives it, uses the destination UDP port to translate the destination IP address back to the IP address of the NAS, and forwards the reply to the corresponding NAS.

The figure below demonstrates how the source IP addresses of several NASs are translated to a single IP address as they pass through the NAT or PAT device on the way to the IP cloud.



RADIUS servers normally check the source IP address in the IP header of the RADIUS packets to track the source of the RADIUS requests and to maintain security. The NAT or PAT solution satisfies these requirements because only a single source IP address is used even though RADIUS packets come from different NAS routers.

However, when retrieving accounting records from the RADIUS database, some billing systems use RADIUS attribute 4, NAS-IP-Address, in the accounting records. The value of this attribute is recorded on the NAS routers as their own IP addresses. The NAS routers are not aware of the NAT or PAT that runs between them and the RADIUS server; therefore, different RADIUS attribute 4 addresses will be recorded in the accounting

records for users from the different NAS routers. These addresses eventually expose different NAS routers to the RADIUS server and to the corresponding billing systems.

Using the RADIUS NAS-IP-Address Attribute Configurability Feature

The RADIUS NAS-IP-Address Attribute Configurability feature allows you to freely configure an arbitrary IP address as RADIUS NAS-IP-Address, RADIUS attribute 4. By manually configuring the same IP address, most likely the IP address on the loopback interface of the NAT or PAT device, for all the routers, you can hide a cluster of NAS routers behind the NAT or PAT device from the RADIUS server.

How to Configure RADIUS NAS-IP-Address Attribute Configurability

Configuring RADIUS NAS-IP-Address Attribute Configurability

Before configuring the RADIUS NAS-IP-Address Attribute Configurability feature, you must have configured the RADIUS servers or server groups and AAA method lists.

To configure the RADIUS NAS-IP-Address Attribute Configurability feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server attribute 4** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute 4** *ip-address* **Example:** `Router (config)# radius-server attribute 4 10.2.1.1`	Configures an IP address to be used as the RADIUS NAS-IP-Address, attribute 4.

Monitoring and Maintaining RADIUS NAS-IP-Address Attribute Configurability

To monitor the RADIUS attribute 4 address that is being used inside the RADIUS packets, use the **debug radius** command.

SUMMARY STEPS

1. **enable**
2. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.

Example

The following sample output is from the **debug radius** command:

```
Router# debug radius
RADIUS/ENCODE(0000001C): acct_session_id: 29
RADIUS(0000001C): sending
RADIUS(0000001C): Send Access-Request to 10.0.0.10:1645 id 21645/17, len 81
RADIUS:  authenticator D0 27 34 C0 F0 C4 1C 1B - 3C 47 08 A2 7E E1 63 2F
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS:  User-Name           [1]   18  "shashi@pepsi.com"
RADIUS:  CHAP-Password       [3]   19  *
RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  NAS-IP-Address      [4]   6   10.0.0.21
UDP: sent src=10.1.1.1(21645), dst=10.0.0.10(1645), length=109
UDP: rcvd src=10.0.0.10(1645), dst=10.1.1.1(21645), length=40
RADIUS: Received from id 21645/17 10.0.0.10:1645, Access-Accept, len 32
RADIUS:  authenticator C6 99 EC 1A 47 0A 5F F2 - B8 30 4A 4C FF 4B 1D F0
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS(0000001C): Received from id 21645/17
```

Configuration Examples for RADIUS NAS-IP-Address Attribute Configurability

Configuring a RADIUS NAS-IP-Address Attribute Configurability Example

The following example shows that IP address 10.0.0.21 has been configured as the RADIUS NAS-IP-Address attribute:

```
radius-server attribute 4 10.0.0.21
radius-server host 10.0.0.10 auth-port 1645 acct-port 1646 key cisco
```

Additional References

The following sections provide references related to RADIUS NAS-IP-Address Attribute Configurability.

Related Documents

Related Topic	Document Title
Configuring AAA	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide: Securing User Services*
Configuring RADIUS	" Configuring RADIUS " module.
RADIUS commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

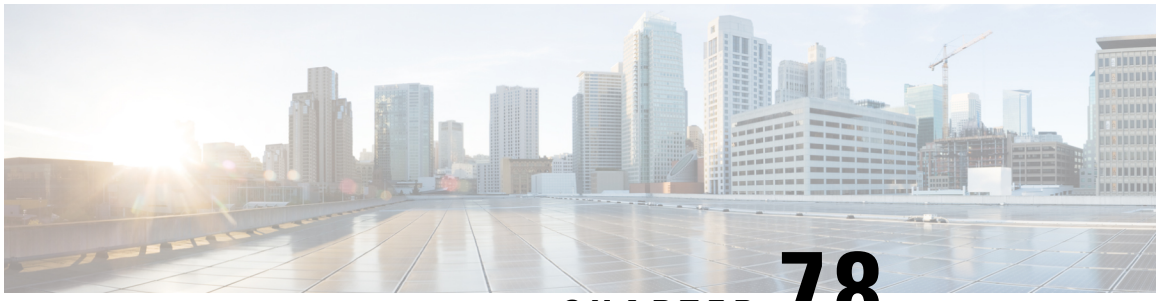
Feature Information for RADIUS NAS-IP-Address Attribute Configurability

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 111: Feature Information for RADIUS NAS-IP-Address Attribute Configurability

Feature Name	Releases	Feature Information
RADIUS NAS-IP-Address Attribute Configurability	Cisco IOS XE Release 3.9S	This feature allows an arbitrary IP address to be configured and used as RADIUS attribute 4, NAS-IP-Address, without changing the source IP address in the IP header of the RADIUS packets. The **radius-server attribute 4** command was introduced this feature.



CHAPTER 78

RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

The RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level feature allows configurations to be customized for different RADIUS server groups. This flexibility allows customized network access server- (NAS-) port formats to be used instead of global formats.

Prerequisites for RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

- You must be running a Cisco IOS image that contains the authentication, authorization, and accounting (AAA) component.

Information About RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

RADIUS Attribute 5 Format Customization

Prior to Cisco IOS Release 12.3(14)T, Cisco IOS software allowed RADIUS attributes that were sent in access requests or accounting requests to be customized on a global basis. You could customize how each configurable

attribute should function when communicating with a RADIUS server. Since the implementation of server groups, global attribute configurations were not flexible enough to address the different customizations that were required to support the various RADIUS servers with which a router might be interacting. For example, if you configured the **global radius-server attribute nas-port format command**option, every service on the router that interacted with a RADIUS server was used in the same way.

Effective with Cisco IOS Release 12.3(14)T, you can configure your router to support override flexibility for per-server groups. You can configure services to use specific named methods for different service types on a RADIUS server. The service types can be set to use their own respective service groups. This flexibility allows customized NAS-port formats to be used instead of the global formats.

How to Configure RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

Configuring the RADIUS Attribute 5 Format on a Per-Server Group Level

To configure your router to support the RADIUS Attribute 5 format on a per-server group level, perform the following steps.

Note To use this per-server group capability, you must actively use a named method list within your services. You can configure one client to use a specific named method while other clients use the default format.

Before you begin

Before performing these steps, you should first configure method lists for AAA as is applicable for your situation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group-name*
4. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
5. **attribute nas-port format** *format-type* [*string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group-name* **Example:** `Router (config)# aaa group server radius radius1`	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 4	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router (server-group)# server 172.101.159.172 auth-port 1645 acct-port 1646`	Configures the IP address of the RADIUS server for the group server.
Step 5	**attribute nas-port format** *format-type* [*string*] **Example:** `Router (server-group)# attribute nas-port format d`	Configures a service to use specific named methods for different service types. • The service types can be set to use their own respective server groups.

Monitoring and Maintaining RADIUS Attribute 5 Format on a Per-Server Group Level

To monitor and maintain RADIUS Attribute 5 Format on a Per-Server Group Level, perform the following steps (the **debug** commands may be used separately):

SUMMARY STEPS

1. **enable**
2. **debug aaa sg-server selection**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**debug aaa sg-server selection** **Example:** `Router# debug aaa sg-server selection`	Displays information about why the RADIUS and TACACS+ server group system in a router is choosing a particular server.
Step 3	**debug radius** **Example:** `Router# debug radius`	Displays information showing that a server group has been selected for a particular request.

Configuration Examples for RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

RADIUS Attribute 5 Format Specified on a Per-Server Level Example

The following configuration example shows a leased-line PPP client that has chosen to send no RADIUS Attribute 5 while the default is to use format F:\tips-migration

```
interface Serial2/0
 no ip address
 encapsulation ppp
 ppp accounting SerialAccounting
 ppp authentication pap
aaa accounting network default start-stop group radius
aaa accounting network SerialAccounting start-stop group group1
aaa group server radius group1
 server 10.101.159.172 auth-port 1645 acct-port 1646
 attribute nas-port none
radius-server host 10.101.159.172 auth-port 1645 acct-port 1646
radius-server attribute nas-port format d
```

Additional References

The following sections provide references related to RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	*Cisco IOS Security Command Reference*
Security Features	*Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2

Related Topic	Document Title
Security Server Protocols	Security Server Protocols section of the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
RADIUS Configuration	Configuring RADIUS feature module.

Standards

Standard	Title
Internet Engineering Task Force (IETF) Internet Draft: Network Access Servers Requirements	Network Access Servers Requirements: Extended RADIUS Practices

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	Remote Authentication Dial In User Service (RADIUS)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for RADIUS Attribute 5 NAS-Port Format Specified on a Per-Server Group Level

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

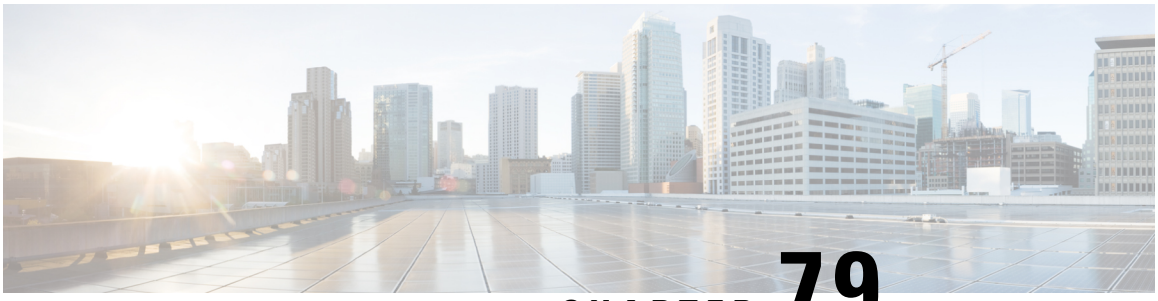
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 112: Feature Information for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

Feature Name	Releases	Feature Information
RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level	Cisco IOS XE Release 3.9S	The RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level feature allows configurations to be customized for different RADIUS server groups. This flexibility allows customized network access server- (NAS-) port formats to be used instead of global formats. The following commands were introduced or modifieF:\tips-migration **attribute nas-port format**.

PART VI

TACACS



CHAPTER 79

Configuring TACACS

This chapter discusses how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

Information About TACACS

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your network access server are available.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service--authentication, authorization, and accounting--independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a methodology for managing multiple network access points from a single management service. The Cisco family of access servers and routers and the Cisco IOS and Cisco IOS XE user interface (for both routers and access servers) can be network access servers.

Network access points enable traditional “dumb” terminals, terminal emulators, workstations, personal computers (PCs), and routers in conjunction with suitable adapters (for example, modems or ISDN adapters) to communicate using protocols such as Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Compressed SLIP (CSLIP), or AppleTalk Remote Access (ARA) protocol. In other words, a network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks. The entities connected to the network through a network access server are called *network access clients* ; for example, a PC running PPP over a voice-grade circuit is a network access client. TACACS+, administered through the AAA security services, can provide the following services:

- Authentication--Provides complete control of authentication through login and password dialog, challenge and response, messaging support.

The authentication facility provides the ability to conduct an arbitrary dialog with the user (for example, after a login and password are provided, to challenge a user with a number of questions, like home address, mother's maiden name, service type, and social security number). In addition, the TACACS+ authentication service supports sending messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.

- Authorization--Provides fine-grained control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user may execute with the TACACS+ authorization feature.
- Accounting--Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the network access server and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between a network access server and a TACACS+ daemon are encrypted.

You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

Cisco makes the TACACS+ protocol specification available as a draft RFC for those customers interested in developing their own TACACS+ software.

TACACS Operation

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following process typically occurs:

1. When the connection is established, the network access server will contact the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

Note TACACS+ allows an arbitrary conversation to be held between the daemon and the user until the daemon receives enough information to authenticate the user. This is usually done by prompting for a username and password combination, but may include other items, such as mother's maiden name, all under the control of the TACACS+ daemon.

1. The network access server will eventually receive one of the following responses from the TACACS+ daemon:
   a. ACCEPT--The user is authenticated and service may begin. If the network access server is configured to requite authorization, authorization will begin at this time.
   b. REJECT--The user has failed to authenticate. The user may be denied further access, or will be prompted to retry the login sequence depending on the TACACS+ daemon.
   c. ERROR--An error occurred at some time during authentication. This can be either at the daemon or in the network connection between the daemon and the network access server. If an ERROR response

is received, the network access server will typically try to use an alternative method for authenticating the user.

d. CONTINUE--The user is prompted for additional authentication information.

2. A PAP login is similar to an ASCII login, except that the username and password arrive at the network access server in a PAP protocol packet instead of being typed in by the user, so the user is not prompted. PPP CHAP logins are also similar in principle.

Following authentication, the user will also be required to undergo an additional authorization phase, if authorization has been enabled on the network access server. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

1. If TACACS+ authorization is required, the TACACS+ daemon is again contacted and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response will contain data in the form of attributes that are used to direct the EXEC or NETWORK session for that user, determining services that the user can access. Services include the following:

   a. Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
   b. Connection parameters, including the host or client IP address, access list, and user timeouts

How to Configure TACACS

To configure your router to support TACACS+, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use TACACS+. For more information about using the **aaa new-model** command, refer to the chapter “AAA Overview.”
- Use the command to specify the IP address of one or more TACACS+ daemons. Use the command to specify an encryption key that will be used to encrypt all exchanges between the network access server and the TACACS+ daemon. This same key must also be configured on the TACACS+ daemon.
- Use the **aaa authentication** global configuration command to define method lists that use TACACS+ for authentication. For more information about using the **aaa authentication** command, refer to the chapter “Configuring Authentication.”
- Use line and interface commands to apply the defined method lists to various interfaces. For more information, refer to the chapter “Configuring Authentication.”
- If needed, use the **aaa authorization** global command to configure authorization for the network access server. Unlike authentication, which can be configured per line or per interface, authorization is configured globally for the entire network access server. For more information about using the **aaa authorization** command, refer to the “Configuring Authorization” chapter.
- If needed, use the **aaa accounting** command to enable accounting for TACACS+ connections. For more information about using the **aaa accounting** command, refer to the “Configuring Accounting” chapter.

Identifying the TACACS Server Host

The command enables you to specify the names of the IP host or hosts maintaining a TACACS+ server. Because the TACACS+ software searches for the hosts in the order specified, this feature can be useful for setting up a list of preferred daemons.

To specify a TACACS+ host, use the following command in global configuration mode:

Command	Purpose
`Router(config)# hostname [single-connection] [port integer] [timeout integer] [key string]`	Specifies a TACACS+ host.

Using the command, you can also configure the following options:

- Use the **single-connection** keyword to specify single-connection. Rather than have the router open and close a TCP connection to the daemon each time it must communicate, the single-connection option maintains a single open connection between the router and the daemon. This is more efficient because it allows the daemon to handle a higher number of TACACS operations.

Note The daemon must support single-connection mode for this to be effective, otherwise the connection between the network access server and the daemon will lock up or you will receive spurious errors.

- Use the **port** *integer* argument to specify the TCP port number to be used when making connections to the TACACS+ daemon. The default port number is 49.
- Use the **timeout** *integer* argument to specify the period of time (in seconds) the router will wait for a response from the daemon before it times out and declares an error.

Note Specifying the timeout value with the command overrides the default timeout value set with the command for this server only.

- Use the **key** *string* argument to specify an encryption key for encrypting and decrypting all traffic between the network access server and the TACACS+ daemon.

Note Specifying the encryption key with the command overrides the default key set by the global configuration command for this server only.

Because some of the parameters of the command override global settings made by the and commands, you can use this command to enhance security on your network by uniquely configuring individual TACACS+ connections.

Setting the TACACS Authentication Key

To set the global TACACS+ authentication key and encryption key, use the following command in global configuration mode:

Command	Purpose
`Router(config)# key`	Sets the encryption key to match that used on the TACACS+ daemon.

Note You must configure the same key on the TACACS+ daemon for encryption to be successful.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups can include multiple host entries as long as each entry has a unique IP address. If two different host entries in the server group are configured for the same service--for example, accounting--the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry for accounting services. (The TACACS+ host entries will be tried in the order in which they are configured.)

To define a server host with a server group name, enter the following commands starting in global configuration mode. The listed server must exist in global configuration mode:

Step 1 Router(config)# *name* [**single-connection**] [**port** *integer*] [**timeout** *integer*] [**key** *string*]

Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the Identifying the TACACS Server Host section of this chapter for more information on the command.

Step 2 Router(config-if)# **aaa group server**{**radius** | **tacacs+**} *group-name*

Defines the AAA server-group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.

Step 3 Router(config-sg)# **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]

Associates a particular TACACS+ server with the defined server group. Use the **auth-port** *port-number* option to configure a specific UDP port solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port solely for accounting.

Repeat this step for each TACACS+ server in the AAA server group.

Note Each server in the group must be defined previously using the command.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS XE software allows you to authenticate users to a particular AAA server group based on the Dialed Number Identification Service (DNIS) number of the session. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different TACACS+ server groups for different customers (that is, different TACACS+ servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS XE software provides the flexibility to implement authentication and accounting services in several ways:

- Globally--AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per interface--AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping--You can use DNIS to specify an AAA server to supply AAA services.

Because AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS--If you configure the network access server to use DNIS to identify which server group provides AAA services, then this method takes precedence over any additional AAA selection method.
- Per interface--If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally--If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the lowest precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the remote security servers associated with each AAA server group. See the Identifying the TACACS Server Host and Configuring AAA Server Groups.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

Step 1 Router(config)# **aaa dnis map enable**

Enables DNIS mapping.

Step 2 Router(config)# **aaa dnis map** *dnis-number* **authentication ppp group** *server-group-name*

Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.

Step 3 Router(config)# **aaa dnis map** *dnis-number* **accounting network** [**none** | **start-stop** | **stop-only**] **group** *server-group-name*

Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Specifying TACACS Authentication

After you have identified the TACACS+ daemon and defined an associated TACACS+ encryption key, you must define method lists for TACACS+ authentication. Because TACACS+ authentication is operated via AAA, you need to issue the **aaa authentication** command, specifying TACACS+ as the authentication method. For more information, refer to the chapter "Configuring Authentication."

Specifying TACACS Authorization

AAA authorization enables you to set parameters that restrict a user's access to the network. Authorization via TACACS+ may be applied to commands, network connections, and EXEC sessions. Because TACACS+ authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying TACACS+ as the authorization method. For more information, refer to the chapter "Configuring Authorization."

Specifying TACACS Accounting

AAA accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because TACACS+ accounting is facilitated through AAA, you must issue the **aaa accounting** command, specifying TACACS+ as the accounting method. For more information, refer to the chapter "Configuring Accounting."

TACACS AV Pairs

The network access server implements TACACS+ authorization and accounting functions by transmitting and receiving TACACS+ attribute-value (AV) pairs for each user session. For a list of supported TACACS+ AV pairs, refer to the TACACS Attribute-Value Pairs chapter.

TACACS Configuration Examples

TACACS Authentication Examples

The following example shows how to configure TACACS+ as the security protocol for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
 10.1.2.3
 key goaway
interface serial 0
 ppp authentication chap pap test
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

- The command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the test method list to this line.

The following example shows how to configure TACACS+ as the security protocol for PPP authentication, but instead of the “test” method list, the “default” method list is used.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
 10.1.2.3
 key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list “MIS-access” instead of “default”:

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
 10.1.2.3
 key goaway
interface serial 0
 ppp authentication pap MIS-access
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “MIS-access,” to be used on serial interfaces running PPP. The method list, “MIS-access,” means that PPP authentication is applied to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

- The command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows the configuration for a TACACS+ daemon with an IP address of 10.2.3.4 and an encryption key of “apple”:

```
aaa new-model
aaa authentication login default group tacacs+ local
 10.2.3.4
 key apple
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines the default method list. Incoming ASCII logins on all interfaces (by default) will use TACACS+ for authentication. If no TACACS+ server responds, then the network access server will use the information contained in the local username database for authentication.
- The command identifies the TACACS+ daemon as having an IP address of 10.2.3.4. The command defines the shared encryption key to be “apple.”

TACACS Authorization Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure network authorization via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
 10.1.2.3
 key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa authorization** command configures network authorization via TACACS+. Unlike authentication lists, this authorization list always applies to all incoming network connections made to the network access server.
- The command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The command defines the shared encryption key to be “goaway.”

- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS Accounting Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure accounting via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
 10.1.2.3
 key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa accounting** command configures network accounting via TACACS+. In this example, accounting records describing the session that just terminated will be sent to the TACACS+ daemon whenever a network connection terminates.
- The command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS Server Group Example

The following example shows how to create a server group with three different TACACS+ servers members:

```
aaa group server tacacs tacgroup1
server 172.16.1.1
server 172.16.1.21
server 172.16.1.31
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select TACAC+ server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the TACACS+ servers that will be associated
! with one of the defined server groups.
 172.16.0.1
 172.17.0.1
 172.18.0.1
 172.19.0.1
 172.20.0.1
 key abcdefg
! The following commands define the sg1 TACACS+ server group and associate servers
! with it.
aaa group server tacacs sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg2
  server 172.18.0.1
! The following commands define the sg3 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg3
  server 172.19.0.1
! The following commands define the default-group TACACS+ server group and associate
! a server with it.
aaa group server tacacs default-group
  server 172.20.0.1
!
! The next set of commands configures default-group tacacs server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using DNIS
! 7777 are sent to the sg1 server group. The accounting records for these connections
! (specifically, start-stop records) are handled by the sg2 server group. Calls with a
! DNIS of 8888 use server group sg3 for authentication and server group default-group
! for accounting. Calls with a DNIS of 9999 use server group default-group for
! authentication and server group sg3 for accounting records (stop records only). All
! other calls with DNIS other than the ones defined use the server group default-group
! for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

TACACS Daemon Configuration Example

The following example shows a sample configuration of the TACACS+ daemon. The precise syntax used by your TACACS+ daemon may be different from what is included in this example.

```
user = mci_customer1 {
 chap = cleartext "some chap password"
 service = ppp protocol = ip {
inacl#1="permit ip any any precedence immediate"
inacl#2="deny igrp 0.0.1.2 255.255.0.0 any"
 }
```

Additional References

The following sections provide references related to the Configuring TACACS+ feature.

Related Documents

Related Topic	Document Title
TACACS+ commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

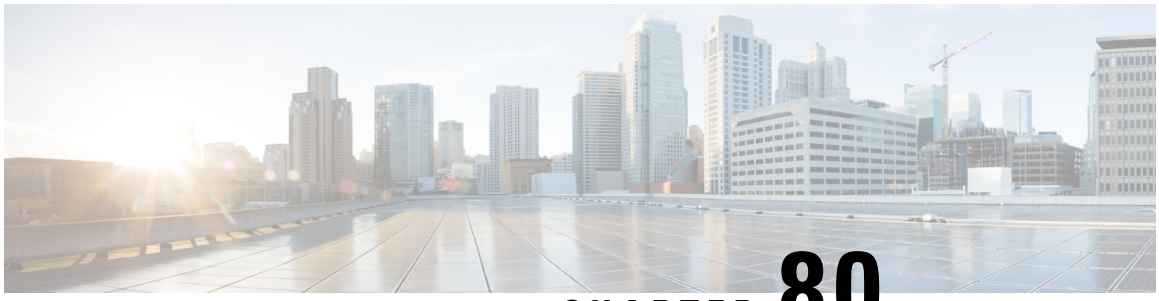
Feature Information for Configuring TACACS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 113: Feature Information for Configuring TACACS+

Feature Name	Releases	Feature Information
TACACS+		TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands. The following commands were introduced or modified: , , **aaa authentication**, **aaa accounting**, **aaa group server tacacs+**.
AAA Server Groups Based on DNIS		The AAA Server Groups Based on DNIS feature allows you to authenticate users to a particular AAA server group based on the Dialed Number Identification Service (DNIS) number of the session. The following commands were introduced or modified: **aaa dnis map enable**, **aaa dnis map authentication group**, **aaa dnis map accounting**.



CHAPTER 80

Per VRF for TACACS Servers

The Per VRF for TACACS+ Servers feature allows per virtual route forwarding (per VRF) to be configured for authentication, authorization, and accounting (AAA) on TACACS+ servers.

Prerequisites for Per VRF for TACACS Servers

- TACACS+ server access is required.
- Experience configuring TACACS+, AAA and per VRF AAA, and group servers is necessary.

Restrictions for Per VRF for TACACS Servers

- The VRF instance must be enabled globally on the router before per VRF for a TACACS+ server is configured.

Information About Per VRF for TACACS Servers

Per VRF for TACACS Servers Overview

The Per VRF for TACACS+ Servers feature allows per VRF AAA to be configured on TACACS+ servers. Prior to Cisco IOS XE Release 2.2, this functionality was available only on RADIUS servers.

How to Configure Per VRF for TACACS Servers

Configuring Per VRF on a TACACS Server

The initial steps in this procedure are used to configure AAA and a server group, create a VRF routing table, and configure an interface. Steps 10 through 13 are used to configure the per VRF on a TACACS+ server feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **interface** *interface-name*
7. **ip vrf forwarding** *vrf-name*
8. **ip address** *ip-address mask* [**secondary**]
9. **exit**
10. **aaa group server tacacs+** *group-name*
11. **server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]
12. **ip vrf forwarding** *vrf-name*
13. **ip tacacs source-interface** *subinterface-name*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router (config)# ip vrf cisco`	Configures a VRF table and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**rd** *route-distinguisher* **Example:** `Router (config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance.
Step 5	**exit** **Example:** `Router (config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Loopback0`	Configures an interface and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-if)# ip vrf forwarding cisco`	Configures a VRF for the interface.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Router (config-if)# ip address 10.0.0.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 9	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 10	**aaa group server tacacs+** *group-name* **Example:** `Router (config)# aaa group server tacacs+ tacacs1`	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Router (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	Configures the IP address of the private TACACS+ server for the group server.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-sg-tacacs+)# ip vrf forwarding cisco`	Configures the VRF reference of a AAA TACACS+ server group.

	Command or Action	Purpose
Step 13	**ip tacacs source-interface** *subinterface-name* **Example:** `Router (config-sg-tacacs+)# ip tacacs source-interface Loopback0`	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	**exit** **Example:** `Router (config-sg-tacacs)# exit`	Exits server-group configuration mode.

Verifying Per VRF for TACACS Servers

To verify the per VRF TACACS+ configuration, perform the following steps:

Note The **debug** commands may be used in any order.

Caution Enabling debug CLI can cause performance degradation on the router. Use of **debug** commands for large number of sessions is not recommended.

SUMMARY STEPS

1. **enable**
2. **debug tacacs authentication**
3. **debug tacacs authorization**
4. **debug tacacs accounting**
5. **debug tacacs packets**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug tacacs authentication** **Example:** `Router# debug tacacs authentication`	Displays information about AAA/TACACS+ authentication.

	Command or Action	Purpose
Step 3	**debug tacacs authorization** **Example:** `Router# debug tacacs authorization`	Displays information about AAA/TACACS+ authorization.
Step 4	**debug tacacs accounting** **Example:** `Router# debug tacacs accounting`	Displays information about accountable events as they occur.
Step 5	**debug tacacs packets** **Example:** `Router# debug tacacs packets`	Displays information about TACACS+ packets.

Configuration Examples for Per VRF for TACACS Servers

Configuring Per VRF for TACACS Servers Example

The following output example shows that the group server **tacacs1** is configured for per VRF AAA services:

```
aaa group server tacacs+ tacacs1
    server-private 10.1.1.1 port 19 key cisco
    ip vrf forwarding cisco
    ip tacacs source-interface Loopback0
  ip vrf cisco
   rd 100:1
  interface Loopback0
   ip address 10.0.0.2 255.0.0.0
   ip vrf forwarding cisco
```

Additional References

The following sections provide references related to Per VRF for TACACS+ Servers..

Related Documents

Related Topic	Document Title
Configuring TACACS+	Configuring TACACS+ module.
Per VRF AAA	Per VRF AAA module.
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

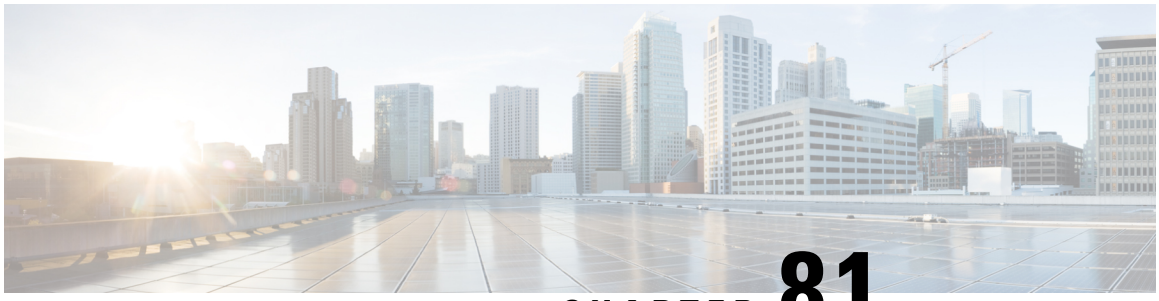
Feature Information for Per VRF for TACACS Servers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 114: Feature Information for Per VRF for TACACS+ Servers

Feature Name	Releases	Feature Information
Per VRF for TACACS+ Servers	Cisco IOS XE Release 2.2	The Per VRF for TACACS+ Servers feature allows per virtual route forwarding (per VRF) to be configured for authentication, authorization, and accounting (AAA) on TACACS+ servers. In Cisco IOS XE Release 2.2, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ip tacacs source-interface**, **ip vrf forwarding (server-group)**, **server-private (TACACS+)**.



CHAPTER 81

TACACS Attribute-Value Pairs

Terminal Access Controller Access Control System Plus (TACACS+) attribute-value (AV) pairs are used to define specific authentication, authorization, and accounting elements in a user profile that is stored on the TACACS+ daemon. This chapter lists the TACACS+ AV pairs currently supported.

Information About TACACS Attribute-Value Pairs

TACACS Authentication and Authorization AV Pairs

The following table lists and describes the supported TACACS+ authentication and authorization AV pairs and specifies the Cisco IOS release in which they are implemented.

Table 115: Supported TACACS+ Authentication and Authorization AV Pairs

Attribute	Description	IOS XE 2.1
acl=x	ASCII number representing a connection access list. Used only when service=shell.	yes
addr=x	A network address. Used with service=slip, service=ppp, and protocol=ip. Contains the IP address that the remote host should use when connecting via SLIP or PPP/IP. For example, addr=10.2.3.4.	yes

Attribute	Description	IOS XE 2.1
addr-pool=x	Specifies the name of a local pool from which to get the address of the remote host. Used with service=ppp and protocol=ip. Note that **addr-pool** works in conjunction with local pooling. It specifies the name of a local pool (which must be preconfigured on the network access server). Use the **ip-local pool** command to declare local pools. For example: ip address-pool local ip local pool boo 10.0.0.1 10.0.0.10 ip local pool moo 10.0.0.1 10.0.0.20 You can then use TACACS+ to return addr-pool=boo or addr-pool=moo to indicate the address pool from which you want to get this remote node's address.	yes
autocmd=x	Specifies an autocommand to be executed at EXEC startup (for example, autocmd=telnet example.com). Used only with service=shell.	yes
callback- dialstring	Sets the telephone number for a callback (for example: callback-dialstring= 408-555-1212). Value is NULL, or a dial-string. A NULL value indicates that the service might choose to get the dial string through other means. Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	yes
callback-line	The number of a TTY line to use for callback (for example: callback-line=4). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	yes
callback-rotary	The number of a rotary group (between 0 and 100 inclusive) to use for callback (for example: callback-rotary=34). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	yes
cmd-arg=x	An argument to a shell (EXEC) command. This indicates an argument for the shell command that is to be run. Multiple cmd-arg attributes can be specified, and they are order dependent. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes
cmd=x	A shell (EXEC) command. This indicates the command name for a shell command that is to be run. This attribute must be specified if service equals "shell." A NULL value indicates that the shell itself is being referred to. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes
data-service	Used with the service=outbound and protocol=ip.	yes
dial-number	Defines the number to dial. Used with the service=outbound and protocol=ip.	yes

Attribute	Description	IOS XE 2.1
dns-servers=	Identifies a DNS server (primary or secondary) that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each DNS server is entered in dotted decimal format.	yes
force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available. To turn on this attribute, use the "true" value (force-56=true). Any other value is treated as false. Used with the service=outbound and protocol=ip.	yes
gw-password	Specifies the password for the home gateway during the L2TP tunnel authentication. Used with service=ppp and protocol=vpdn.	yes
idletime=x	Sets a value, in minutes, after which an idle session is terminated. A value of zero indicates no timeout.	yes
inacl#<n>	ASCII access list identifier for an input access list to be installed and applied to an interface for the duration of the current connection. Used with service=ppp and protocol=ip, and service service=ppp and protocol =ipx. Per-user access lists do not currently work with ISDN interfaces.	yes
inacl=x	ASCII identifier for an interface input access list. Used with service=ppp and protocol=ip. Per-user access lists do not currently work with ISDN interfaces.	yes
interface-config#<n>	Specifies user-specific AAA interface configuration information with Virtual Profiles. The information that follows the equal sign (=) can be any Cisco IOS interface configuration command. Multiple instances of the attributes are allowed, but each instance must have a unique number. Used with service=ppp and protocol=lcp. **Note** This attribute replaces the "interface-config=" attribute.	yes
ip-addresses	Space-separated list of possible IP addresses that can be used for the end-point of a tunnel. Used with service=ppp and protocol=vpdn.	yes
l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template. Used with service=ppp and protocol=vpdn.	yes
l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment. Used with service=ppp and protocol=vpdn.	yes
l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received. Used with service=ppp and protocol=vpdn.	yes

Attribute	Description	IOS XE 2.1
l2tp-hello- interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here. Used with service=ppp and protocol=vpdn.	yes
l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden. Used with service=ppp and protocol=vpdn.	yes
l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down. Used with service=ppp and protocol=vpdn.	yes
l2tp-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS. Used with service=ppp and protocol=vpdn.	yes
l2tp-tunnel- authen	If this attribute is set, it performs L2TP tunnel authentication. Used with service=ppp and protocol=vpdn.	yes
l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding. Used with service=ppp and protocol=vpdn.	yes
l2tp-udp- checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no. Used with service=ppp and protocol=vpdn.	yes
link- compression=	Defines whether to turn on or turn off “stac” compression over a PPP link. Used with service=ppp. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac	yes
load-threshold= <n>	Sets the load threshold for the caller at which additional links are either added to or deleted from the multilink bundle. If the load goes above the specified value, additional links are added. If the load goes below the specified value, links are deleted. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	yes
map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out. Used with the service=outbound and protocol=ip.	yes
max-links=<n>	Restricts the number of links that a user can have in a multilink bundle. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	yes

Attribute	Description	IOS XE 2.1
min-links	Sets the minimum number of links for MLP. Used with service=ppp and protocol=multilink, protocol=vpdn.	yes
nas-password	Specifies the password for the network access server during the L2TP tunnel authentication. Used with service=ppp and protocol=vpdn.	yes
nocallback-verify	Indicates that no callback verification is required. The only valid value for this parameter is 1 (for example, nocallback-verify=1). Used with service=arap, service=slip, service=ppp, service=shell. There is no authentication on callback. Not valid for ISDN.	yes
noescape=x	Prevents user from using an escape character. Used with service=shell. Can be either true or false (for example, noescape=true).	yes
nohangup=x	Used with service=shell. Specifies the nohangup option, which means that after an EXEC shell is terminated, the user is presented with another login (username) prompt. Can be either true or false (for example, nohangup=false).	yes
old-prompts	Allows providers to make the prompts in TACACS+ appear identical to those of earlier systems (TACACS and Extended TACACS). This allows administrators to upgrade from TACACS or Extended TACACS to TACACS+ transparently to users.	yes
outacl#<n>	ASCII access list identifier for an interface output access list to be installed and applied to an interface for the duration of the current condition. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Per-user access lists do not currently work with ISDN interfaces.	yes
outacl=x	ASCII identifier for an interface output access list. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Contains an IP output access list for SLIP or PPP/IP (for example, outacl=4). The access list itself must be preconfigured on the router. Per-user access lists do not currently work with ISDN interfaces.	yes
pool-def#<n>	Defines IP address pools on the network access server. Used with service=ppp and protocol=ip.	yes
pool-timeout=	Defines (in conjunction with pool-def) IP address pools on the network access server. During IPCP address negotiation, if an IP pool name is specified for a user (see the addr-pool attribute), a check is made to see if the named pool is defined on the network access server. If it is, the pool is consulted for an IP address. Used with service=ppp and protocol=ip.	yes

Attribute	Description	IOS XE 2.1
port-type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN- Asynchronous (V.110) • 5: Virtual Used with service=any and protocol=aaa.	yes
ppp-vj-slot-compression	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.	yes
priv-lvl=x	Privilege level to be assigned for the EXEC. Used with service=shell. Privilege levels range from 0 to 15, with 15 being the highest.	yes
protocol=x	A protocol that is a subset of a service. An example would be any PPP NCP. Currently known values are **lcp**, **ip**, **ipx**, **atalk**, **vines**, **lat**, **xremote**, **tn3270**, **telnet**, **rlogin**, **pad**, **vpdn**, **osicp**, **deccp**, **ccp**, **cdp**, **bridging**, **xns**, **nbf**, **bap**, **multilink**, and **unknown**.	yes
proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces. Used with the service=shell and protocol=exec.	yes
route	Specifies a route to be applied to an interface. Used with service=slip, service=ppp, and protocol=ip. During network authorization, the route attribute can be used to specify a per-user static route, to be installed by TACACS+ as follows: route=" *dst_address mask* [*gateway*]" This indicates a temporary static route that is to be applied. The *dst_address* , *mask* , and *gateway* are expected to be in the usual dotted-decimal notation, with the same meanings as in the familiar **ip route** configuration command on a network access server. If *gateway* is omitted, the peer's address is the gateway. The route is expunged when the connection terminates.	yes
route#<n>	Like the route AV pair, this specifies a route to be applied to an interface, but these routes are numbered, allowing multiple routes to be applied. Used with service=ppp and protocol=ip, and service=ppp and protocol=ipx.	yes

Attribute	Description	IOS XE 2.1
routing=x	Specifies whether routing information is to be propagated to and accepted from this interface. Used with service=slip, service=ppp, and protocol=ip. Equivalent in function to the /routing flag in SLIP and PPP commands. Can either be true or false (for example, routing=true).	yes
rte-fltr-in#<n>	Specifies an input access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	yes
rte-fltr-out#<n>	Specifies an output access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	yes
sap#<n>	Specifies static Service Advertising Protocol (SAP) entries to be installed for the duration of a connection. Used with service=ppp and protocol=ipx.	yes
sap-fltr-in#<n>	Specifies an input SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	yes
sap-fltr-out#<n>	Specifies an output SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	yes
send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication. Used with service=any and protocol=aaa.	yes
send-secret	Specifies the password that the NAS needs to respond to a chap/pap request from the remote end of a connection on an outgoing call. Used with service=ppp and protocol=ip.	yes
service=x	The primary service. Specifying a service attribute indicates that this is a request for authorization or accounting of that service. Current values are **slip**, **ppp**, **arap**, **shell**, **tty-daemon**, **connection**, and **system**. This attribute must always be included.	yes
source-ip=x	Used as the source IP address of all VPDN packets generated as part of a VPDN tunnel. This is equivalent to the Cisco **vpdn outgoing** global configuration command.	yes
spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range. Used with the service=mobileip and protocol=ip.	yes

Attribute	Description	IOS XE 2.1
timeout=x	The number of minutes before an EXEC or ARA session disconnects (for example, timeout=60). A value of zero indicates no timeout. Used with service=arap.	yes
tunnel-id	Specifies the username that will be used to authenticate the tunnel over which the individual user MID will be projected. This is analogous to the *remote name* in the **vpdn outgoing** command. Used with service=ppp and protocol=vpdn.	yes
wins-servers=	Identifies a Windows NT server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each Windows NT server is entered in dotted decimal format.	yes
zonelist=x	A numeric zonelist value. Used with service=arap. Specifies an AppleTalk zonelist for ARA (for example, zonelist=5).	yes

For more information about configuring TACACS+, refer to the chapter "Configuring TACACS+." For more information about configuring TACACS+ authentication and authorization, refer to the chapters "Configuring Authentication" and "Configuring Authorization."

TACACS Accounting AV Pairs

The following table lists and describes the supported TACACS+ accounting AV pairs and specifies the Cisco IOS XE release in which they are implemented.

Table 116: Supported TACACS+ Accounting AV Pairs

Attribute	Description	IOS XE 2.1
Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.	yes
bytes_in	The number of input bytes transferred during this connection.	yes
bytes_out	The number of output bytes transferred during this connection.	yes
Call-Type	Describes the type of fax activity: fax receive or fax send.	yes
cmd	The command the user executed.	yes
data-rate	This AV pair has been renamed. See nas-rx-speed.	

Attribute	Description	IOS XE 2.1
disc-cause	Specifies the reason a connection was taken off-line. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. Refer to the following table (Disconnect Cause Extensions) for a list of Disconnect-Cause values and their meanings.	yes
disc-cause-ext	Extends the disc-cause attribute to support vendor-specific reasons why a connection was taken off-line.	yes
elapsed_time	The elapsed time in seconds for the action. Useful when the device does not keep real time.	yes
Email-Server- Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.	yes
Email-Server-Ack- Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.	yes
event	Information included in the accounting packet that describes a state change in the router. Events described are accounting starting and accounting stopping.	yes
Fax-Account-Id- Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** command.	yes
Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.	yes
Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.	yes
Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.	yes
Fax-Dsn-Address	Indicates the address to which DSNs will be sent.	yes
Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.	yes
Fax-Mdn-Address	Indicates the address to which MDNs will be sent.	yes
Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.	yes

Attribute	Description	IOS XE 2.1
Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.	yes
Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.	yes
Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.	yes
Fax-Process-Abort- Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.	yes
Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.	yes
Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name	yes
mlp-links-max	Gives the count of links which are known to have been in a given multilink session at the time the accounting record is generated.	yes
mlp-sess-id	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. This attribute is sent in authentication-response packets.	yes
nas-rx-speed	Specifies the average number of bits per second over the course of the connection's lifetime. This attribute is sent in accounting-stop records.	yes
nas-tx-speed	Reports the transmit speed negotiated by the two modems.	yes
paks_in	The number of input packets transferred during this connection.	yes
paks_out	The number of output packets transferred during this connection.	yes
port	The port the user was logged in to.	yes
Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.	yes
pre-bytes-in	Records the number of input bytes before authentication. This attribute is sent in accounting-stop records.	yes
pre-bytes-out	Records the number of output bytes before authentication. This attribute is sent in accounting-stop records.	yes
pre-paks-in	Records the number of input packets before authentication. This attribute is sent in accounting-stop records.	yes

Attribute	Description	IOS XE 2.1
pre-paks-out	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.	yes
pre-session-time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication.	yes
priv_level	The privilege level associated with the action.	yes
protocol	The protocol associated with the action.	yes
reason	Information included in the accounting packet that describes the event that caused a system change. Events described are system reload, system shutdown, or when accounting is reconfigured (turned on or off).	yes
service	The service the user used.	yes
start_time	The time the action started (in seconds since the epoch, 12:00 a.m. Jan 1 1970). The clock must be configured to receive this information.	yes
stop_time	The time the action stopped (in seconds since the epoch.) The clock must be configured to receive this information.	yes
task_id	Start and stop records for the same event must have matching (unique) task_id numbers.	yes
timezone	The time zone abbreviation for all timestamps included in this packet.	yes
xmit-rate	This AV pair has been renamed. See nas-tx-speed.	

The following table lists the cause codes and descriptions for the Disconnect Cause Extended (disc-cause-ext) attribute.

Table 117: Disconnect Cause Extensions

Cause Codes	Description	IOS XE 2.1
1000 - No Reason	No reason for the disconnect.	yes
1001 - No Disconnect	The event was not a disconnect.	yes
1002 - Unknown	The reason for the disconnect is unknown. This code can appear when the remote connection goes down.	yes
1003 - Call Disconnect	The call has disconnected.	yes
1004 - CLID Auth Fail	Calling line ID (CLID) authentication has failed.	yes
1009 - No Modem Available	The modem is not available.	yes
1010 - No Carrier	The modem never detected data carrier detect (DCD). This code can appear if a disconnect occurs during the initial modem connection.	yes

Cause Codes	Description	IOS XE 2.1
1011 - Lost Carrier	The modem detected DCD but became inactive. This code can appear if a disconnect occurs during the initial modem connection.	yes
1012 - No Modem Results	The result codes could not be parsed. This code can appear if a disconnect occurs during the initial modem connection.	yes
1020 - TS User Exit	The user exited normally from the terminal server. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1021 - Idle Timeout	The user exited from the terminal server because the idle timer expired. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1022 - TS Exit Telnet	The user exited normally from a Telnet session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1023 - TS No IP Addr	The user could not switch to Serial Line Internet Protocol (SLIP) or PPP because the remote host had no IP address or because the dynamic pool could not assign one. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1024 - TS TCP Raw Exit	The user exited normally from a raw TCP session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1025 - TS Bad Password	The login process ended because the user failed to enter a correct password after three attempts. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1026 - TS No TCP Raw	The raw TCP option is not enabled. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1027 - TS CNTL-C	The login process ended because the user typed Ctrl-C. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1028 - TS Session End	The terminal server session has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1029 - TS Close Vconn	The user closed the virtual connection. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1030 - TS End Vconn	The virtual connection has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes

Cause Codes	Description	IOS XE 2.1
1031 - TS Rlogin Exit	The user exited normally from an Rlogin session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1032 - TS Rlogin Opt Invalid	The user selected an invalid Rlogin option. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1033 - TS Insuff Resources	The access server has insufficient resources for the terminal server session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	yes
1040 - PPP LCP Timeout	PPP link control protocol (LCP) negotiation timed out while waiting for a response from a peer. This code concerns PPP connections.	yes
1041 - PPP LCP Fail	There was a failure to converge on PPP LCP negotiations. This code concerns PPP connections.	yes
1042 - PPP Pap Fail	PPP Password Authentication Protocol (PAP) authentication failed. This code concerns PPP connections.	yes
1043 - PPP CHAP Fail	PPP Challenge Handshake Authentication Protocol (CHAP) authentication failed. This code concerns PPP connections.	yes
1044 - PPP Remote Fail	Authentication failed from the remote server. This code concerns PPP sessions.	yes
1045 - PPP Receive Term	The peer sent a PPP termination request. This code concerns PPP connections.	yes
PPP LCP Close (1046)	LCP got a close request from the upper layer while LCP was in an open state. This code concerns PPP connections.	yes
1047 - PPP No NCP	LCP closed because no NCPs were open. This code concerns PPP connections.	yes
1048 - PPP MP Error	LCP closed because it could not determine to which Multilink PPP bundle that it should add the user. This code concerns PPP connections.	yes
1049 - PPP Max Channels	LCP closed because the access server could not add any more channels to an MP session. This code concerns PPP connections.	yes
1050 - TS Tables Full	The raw TCP or Telnet internal session tables are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	yes
1051 - TS Resource Full	Internal resources are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	yes

Cause Codes	Description	IOS XE 2.1
1052 - TS Invalid IP Addr	The IP address for the Telnet host is invalid. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	yes
1053 - TS Bad Hostname	The access server could not resolve the host name. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	yes
1054 - TS Bad Port	The access server detected a bad or missing port number. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	yes
1060 - TCP Reset	The host reset the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1061 - TCP Connection Refused	The host refused the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1062 - TCP Timeout	The TCP connection timed out. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1063 - TCP Foreign Host Close	A foreign host closed the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1064 - TCP Net Unreachable	The TCP network was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1065 - TCP Host Unreachable	The TCP host was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1066 - TCP Net Admin Unreachable	The TCP network was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1067 - TCP Host Admin Unreachable	The TCP host was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1068 - TCP Port Unreachable	The TCP port was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	yes
1100 - Session Timeout	The session timed out because there was no activity on a PPP link. This code applies to all session types.	yes
1101 - Security Fail	The session failed for security reasons. This code applies to all session types.	yes

Cause Codes	Description	IOS XE 2.1
1102 - Callback	The session ended for callback. This code applies to all session types.	yes
1120 - Unsupported	One end refused the call because the protocol was disabled or unsupported. This code applies to all session types.	yes
1150 - Radius Disc	The RADIUS server requested the disconnect.	yes
1151 - Local Admin Disc	The local administrator has disconnected.	yes
1152 - SNMP Disc	Simple Network Management Protocol (SNMP) has disconnected.	yes
1160 - V110 Retries	The allowed retries for V110 synchronization have been exceeded.	yes
1170 - PPP Auth Timeout	Authentication timeout. This code applies to PPP sessions.	yes
1180 - Local Hangup	The call disconnected as the result of a local hangup.	yes
1185 - Remote Hangup	The call disconnected because the remote end hung up.	yes
1190 - T1 Quiesced	The call disconnected because the T1 line that carried it was quiesced.	yes
1195 - Call Duration	The call disconnected because the call duration exceeded the maximum amount of time allowed by the Max Call Mins or Max DS0 Mins parameter on the access server.	yes
1600 - VPDN User Disconnect	The user disconnected. This value applies to virtual private dial-up network (VPDN) sessions.	yes
1601 - VPDN Carrier Loss	Carrier loss has occurred. This code applies to VPDN sessions.	yes
1602 - VPDN No Resources	There are no resources. This code applies to VPDN sessions.	yes
1603 - VPDN Bad Control Packet	The control packet is invalid. This code applies to VPDN sessions.	yes
1604 - VPDN Admin Disconnect	The administrator disconnected. This code applies to VPDN sessions.	yes
1605 - VPDN Tunnel Down/Setup Fail	The tunnel is down or the setup failed. This code applies to VPDN sessions.	yes
1606 - VPDN Local PPP Disconnect	There was a local PPP disconnect. This code applies to VPDN sessions.	yes
1607 - VPDN Softshut/Session Limit	New sessions cannot be established on the VPN tunnel. This code applies to VPDN sessions.	yes
1608 - VPDN Call Redirected	The call was redirected. This code applies to VPDN sessions.	yes

Cause Codes	Description	IOS XE 2.1
1801 - Q850 Unassigned Number	The number has not been assigned. This code applies to ISDN or modem calls that came in over ISDN.	no
1802 - Q850 No Route	The equipment that is sending this code has received a request to route the call through a particular transit network that it does not recognize. The equipment that is sending this code does not recognize the transit network because either the transit network does not exist or because that particular transit network, while it does exist, does not serve the equipment that is sending this code. This code applies to ISDN or modem calls that came in over ISDN.	no
1803 - Q850 No Route To Destination	The called party cannot be reached because the network through which the call has been routed does not serve the destination that is desired. This code applies to ISDN or modem calls that came in over ISDN.	no
1806 - Q850 Channel Unacceptable	The channel that has been most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that came in over ISDN.	no
1816 - Q850 Normal Clearing	The call is being cleared because one of the users who is involved in the call has requested that the call be cleared. This code applies to ISDN or modem calls that came in over ISDN.	no
1817 - Q850 User Busy	The called party is unable to accept another call because the user-busy condition has been encountered. This code may be generated by the called user or by the network. In the case of the user, the user equipment is compatible with the call. This code applies to ISDN or modem calls that came in over ISDN.	no
1818 - Q850 No User Responding	Used when a called party does not respond to a call-establishment message with either an alerting or connect indication within the prescribed period of time that was allocated. This code applies to ISDN or modem calls that came in over ISDN.	no
1819 - Q850 No User Answer	The called party has been alerted but does not respond with a connect indication within a prescribed period of time. This code applies to ISDN or modem calls that came in over ISDN.	no
1821 - Q850 Call Rejected	The equipment that is sending this code does not wish to accept this call although it could have accepted the call because the equipment that is sending this code is neither busy nor incompatible. This code may also be generated by the network, indicating that the call was cleared due to a supplementary service constraint. The diagnostic field may contain additional information about the supplementary service and reason for rejection. This code applies to ISDN or modem calls that came in over ISDN.	no

Cause Codes	Description	IOS XE 2.1
1822 - Q850 Number Changed	The number that is indicated for the called party is no longer assigned. The new called party number may optionally be included in the diagnostic field. This code applies to ISDN or modem calls that came in over ISDN.	no
1827 - Q850 Destination Out of Order	The destination that was indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term “not functioning correctly” indicates that a signaling message was unable to be delivered to the remote party. This code applies to ISDN or modem calls that came in over ISDN.	no
1828 - Q850 Invalid Number Format	The called party cannot be reached because the called party number is not in a valid format or is not complete. This code applies to ISDN or modem calls that came in over ISDN.	no
1829 - Q850 Facility Rejected	This code is returned when a supplementary service that was requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no
1830 - Q850 Responding to Status Enquiry	This code is included in the STATUS message when the reason for generating the STATUS message was the prior receipt of a STATUS ENQUIRY message. This code applies to ISDN or modem calls that came in over ISDN.	no
1831 - Q850 Unspecified Cause	No other code applies. This code applies to ISDN or modem calls that came in over ISDN.	no
1834 - Q850 No Circuit Available	No circuit or channel is available to handle the call. This code applies to ISDN or modem calls that came in over ISDN.	no
1838 - Q850 Network Out of Order	The network is not functioning correctly and the condition is likely to last a relatively long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no
1841 - Q850 Temporary Failure	The network is not functioning correctly and the condition is not likely to last a long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no
1842 - Q850 Network Congestion	The network is congested. This code applies to ISDN or modem calls that came in over ISDN.	no
1843 - Q850 Access Info Discarded	This code indicates that the network could not deliver access information to the remote user as requested. This code applies to ISDN or modem calls that came in over ISDN.	no
1844 - Q850 Requested Channel Not Available	This code is returned when the circuit or channel that is indicated by the requesting entity cannot be provided by the other side of the interface. This code applies to ISDN or modem calls that came in over ISDN.	no
1845 - Q850 Call Pre-empted	The call was preempted. This code applies to ISDN or modem calls that came in over ISDN.	no

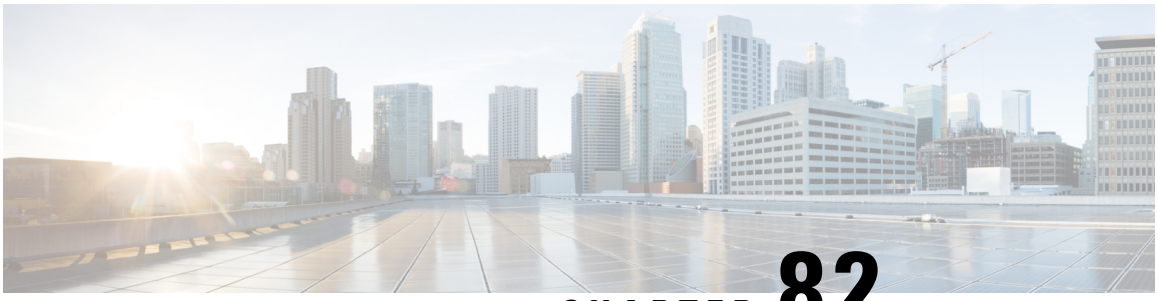
Cause Codes	Description	IOS XE 2.1
1847 - Q850 Resource Unavailable	This code is used to report a resource-unavailable event only when no other code in the resource-unavailable class applies. This code applies to ISDN or modem calls that came in over ISDN.	no
1850 - Q850 Facility Not Subscribed	Not a subscribed facility. This code applies to ISDN or modem calls that came in over ISDN.	no
1852 - Q850 Outgoing Call Barred	Although the calling party is a member of the closed user group for the outgoing closed user group call, outgoing calls are not allowed for this member. This code applies to ISDN or modem calls that came in over ISDN.	no
Q850 Incoming Call Barred (1854)	Although the called party is a member of the closed user group for the incoming closed user group call, incoming calls are not allowed to this member. This code applies to ISDN or modem calls that have come in over ISDN.	no
1858 - Q850 Bearer Capability Not Available	The user has requested a bearer capability that is implemented by the equipment that generated this code but that is not available at this time. This code applies to ISDN or modem calls that have come in over ISDN.	no
1863 - Q850 Service Not Available	The code is used to report a service- or option-not-available event only when no other code in the service- or option-not-available class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no
1865 - Q850 Bearer Capability Not Implemented	The equipment that is sending this code does not support the bearer capability that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no
1866 - Q850 Channel Not Implemented	The equipment that is sending this code does not support the channel type that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no
1869 - Q850 Facility Not Implemented	The supplementary service requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no
1881 - Q850 Invalid Call Reference	The equipment that is sending this code has received a message having a call reference that is not currently in use on the user-network interface. This code applies to ISDN or modem calls that have come in over ISDN.	no
1882 - Q850 Channel Does Not Exist	The channel most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that have come in over ISDN. This code applies to ISDN or modem calls that have come in over ISDN.	no

Cause Codes	Description	IOS XE 2.1
1888 - Q850 Incompatible Destination	The equipment that is sending this code has received a request to establish a call that has low-layer compatibility or other compatibility attributes that cannot be accommodated. This code applies to ISDN or modem calls that have come in over ISDN.	no
1896 - Q850 Mandatory Info Element Is Missing	The equipment that is sending this code has received a message that is missing an information element that must be present in the message before that message can be processed. This code applies to ISDN or modem calls that have come in over ISDN.	no
1897 - Q850 Non Existent Message Type	The equipment that is sending this code has received a message with a message type that it does not recognize either because this is a message that is not defined or that is defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no
1898 - Q850 Invalid Message	This code is used to report an invalid message when no other code in the invalid message class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no
1899 - Q850 Bad Info Element	The information element not recognized. This code applies to ISDN or modem calls that have come in over ISDN.	no
1900 - Q850 Invalid Element Contents	The equipment that is sending this code has received an information element that it has implemented; however, one or more fields in the information element are coded in such a way that has not been implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no
1901 - Q850 Wrong Message for State	The message that was received is incompatible with the call state. This code applies to ISDN or modem calls that have come in over ISDN.	no
1902 - Q850 Recovery on Timer Expiration	A procedure has been initiated by the expiration of a timer in association with error-handling procedures. This code applies to ISDN or modem calls that have come in over ISDN.	no
1903 - Q850 Info Element Error	The equipment that is sending this code has received a message that includes information elements or parameters that are not recognized because the information element identifiers or paramenter names are not defined or are defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no
1911 - Q850 Protocol Error	This code is used to report a protocol error event only when no other code in the protocol error class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no
1927 - Q850 Unspecified Internetworking Event	There has been an error when interworking with a network that does not provide codes for actions that it takes. This code applies to ISDN or modem calls that have come in over ISDN.	no

For more information about configuring TACACS+ accounting, see the Configuring TACACS+ feature module.

PART VII

Cisco TrustSec



CHAPTER 82

Overview of Cisco TrustSec

Cisco TrustSec uses tags to represent logical group privilege. This tag, called a Security Group Tag (SGT), is used in access policies. The SGT is understood and is used to enforce traffic by Cisco switches, routers and firewalls. Cisco TrustSec is defined in three phases: classification, propagation and enforcement.

When users and devices connect to a network, the network assigns a specific security group. This process is called classification. Classification can be based on the results of the authentication or by associating the SGT with an IP, VLAN, or port-profile.

After user traffic is classified, then the SGT is propagated from where classification took place, to where enforcement action is invoked. This process is called propagation. Cisco TrustSec has two methods of SGT propagation: inline tagging and SXP.

With inline tagging, the SGT is embedded into the ethernet frame. The ability to embed the SGT within an ethernet frame does require specific hardware support. Therefore network devices that do not have the hardware support use a protocol called SXP (SGT Exchange Protocol). SXP is used to share the SGT to IP address mapping. This allows the SGT propagation to continue to the next device in the path.

Finally an enforcement device controls traffic based on the tag information. A TrustSec enforcement point can be a Cisco firewall, router, or switch. The enforcement device takes the source SGT and looks it up against the destination SGT to determine if the traffic should be allowed or denied. If the enforcement device is a Cisco firewall, then it also allows stateful firewall processing and IPS deep packet inspection using the same source SGT in a single firewall rule.

Note Cisco TrustSec features are not supported on switch ports on the Cisco 1000 Series Integrated Services Routers.

Note When CTS enforcement is enabled, the device attempts to download policies from ISE and this requires that a RADIUS server is configured. If the RADIUS server is not configured then the policies cannot be downloaded and the Syslog file records the error.

For more information about classification and enforcement, refer to Cisco TrustSec Quick Start Configuration Guide.

SGT Inline Tagging

Each security group in a CTS domain is assigned a unique 16-bit tag called the "Scalable Group Tag" (SGT). The SGT is a single label indicating the privileges of the source within the entire network. It is in turn propagated between network hops allowing any intermediary devices (switches, routers) to enforce polices based on the identity tag.

CTS-capable devices have built-in hardware capabilities than can send and receive packets with SGT embedded in the MAC (L2) layer. This feature is called "L2-SGT Imposition." It allows Ethernet interfaces on the device to be enabled for L2-SGT imposition so that device can insert an SGT in the packet to be carried to its next hop Ethernet neighbor. SGT-over-Ethernet is a method of hop-by-hop propagation of SGT embedded in clear-text (unencrypted) Ethernet packets. Inline identity propagation is scalable, provides near line-rate performance and avoids control plane overhead.

The Cisco TrustSec with SXPv4 feature supports CTS Meta Data (CMD) based L2-SGT. When a packet enters a CTS enabled interface, the IP-SGT mapping database (with dynamic entries built by SXP and/or static entries built by configuration commands) is analyzed to learn the SGT corresponding to the source IP address of the packet, which is then inserted into the packet and carried throughout the network within the CTS header.

As the tag represents the group of the source, the tag is also referred to as the Source Group Tag (SGT). At the egress edge of the network, the group assigned to the packet's destination becomes known. At this point, the access control can be applied. With CTS, access control policies are defined between the security groups and are referred to as Security Group Access Control Lists (SGACL). From the view of any given packet, it is simply being sourced from a security group and destined for another security group.

Protected Access Credential (PAC)

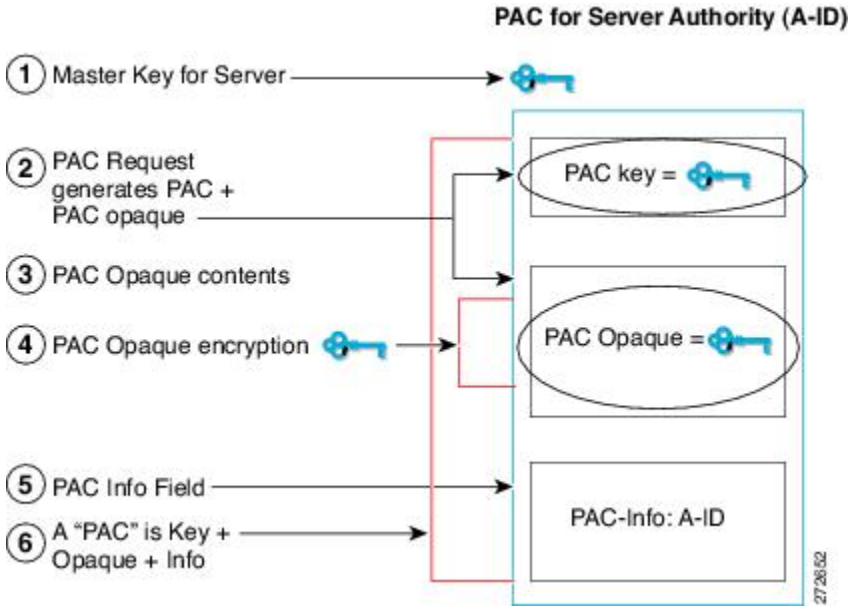
The PAC is a unique shared credential used to mutually authenticate client and server. It is associated with a specific client username and a server authority identifier (A-ID). A PAC removes the need for Public Key Infrastructure (PKI) and digital certificates.

Creating a PAC consists of the following steps:

1. Server A-ID maintains a local key (master key) that is only known by the server.
2. When a client, which is referred to in this context as an initiator identity (I-ID), requests a PAC from the server, the server generates a randomly unique PAC key and PAC-Opaque field for this client.
3. The PAC-Opaque field contains the randomly generated PAC key along with other information such as an I-ID and key lifetime.
4. PAC Key, I-ID, and Lifetime in the PAC-Opaque field are encrypted with the master key.
5. A PAC-Info field that contains the A-ID is created.
6. The PAC is distributed or imported to the client automatically.

Note The server does not maintain the PAC or the PAC key, enabling the EAP-FAST server to be stateless.

The figure below describes the PAC's construction. A PAC consists of the PAC-Opaque, PAC Key, and PAC-Info fields. The PAC-Info field contains the A-ID.



PAC Provisioning

In Secure RADIUS, the PAC key is provisioned into each device during authentication to derive the shared secret. Since the RADIUS ACS does not store the PAC key for each device, the clients must also send an additional RADIUS attribute containing the PAC-Opaque field, which is a variable length field that can only be interpreted by the server to recover the required information and validate the peer's identity and authentication. For example, the PAC-Opaque field may include the PAC key and the PAC's peer identity.

The PAC-Opaque field format and contents are specific to the PAC server on which it is issued. The RADIUS server obtains the PAC Key from the PAC-Opaque field and derives the shared secret the same way clients do. Secure RADIUS only modifies the way shared secret is derived and not its usage.

EAP-FAST Phase 0 is used to automatically provision a client with a PAC.

Deploying Devices in High Availability Setup

Perform the following steps when deploying devices in an HA setup:

1. Clear the credentials from all the devices which are part of the HA setup.
2. Boot the stack setup and establish the device roles (active, standby, and members).
3. Configure the credentials on the active device. Use the **cts credentials id** *id* **password** *password* command to configure the credentials.

Note While adding a new device to an existing stack, ensure that you clear the credentials on the fresh device and then add it to the existing stack setup.

CTS Credentials

CTS requires each device in the network to identity itself uniquely. For use in TrustSec Network Device Admission Control (NDAC) authentication, use the **cts credentials** command to specify the Cisco TrustSec device ID and password for this device to use when authenticating with other Cisco TrustSec devices and for provisioning the PAC (Protected Access Credentials) with EAP-FAST. The CTS credentials state retrieval is not performed by the nonvolatile generation process (NVGEN) because the CTS credential information is saved in the keystore, not in the startup-config. Those credentials are stored in the keystore, eliminating the need to save the running-config. To display the CTS device ID, use the **show cts credentials** command. The stored password is never displayed.

To change the device ID or the password, reenter the command. To clear the keystore, use the **clear cts credentials** command.

Note When the CTS device ID is changed, all Protected Access Credentials (PACs) are flushed from the keystore because the PACs are associated with the old device ID and are not valid for a new identity.

Configuring SGT Inline Tagging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**gigabitethernet** *port* | **vlan** *number*}
4. **cts manual**
5. **policy static sgt** *tag* [**trusted**]
6. **end**
7. **show cts interface brief**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**gigabitethernet** *port* | **vlan** *number*} **Example:** `Device(config)# interface gigabitethernet 0`	Enters the interface on which CTS SGT authorization and forwarding is enabled.
Step 4	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables the interface for CTS SGT authorization and forwarding. Enters CTS manual interface configuration mode. **Note** If you are using subinterfaces, configure the **cts manual** command in the **config-subif** mode (subinterface) instead of the **config-if** mode (parent interface).
Step 5	**policy static sgt** *tag* [**trusted**] **Example:** `Device(config-if-cts-manual)# policy static sgt 77`	Configures a static SGT ingress policy on the interface and defines the trustworthiness of an SGT received on the interface. **Note** The **trusted** keyword indicates that the interface is trustworthy for CTS. The SGT value received in the Ethernet packet on this interface is trusted and will be used by the device for any SG-aware policy enforcement or for purpose of egress-tagging.
Step 6	**end** **Example:** `Device(config-if-cts-manual)# end`	Exits CTS manual interface configuration mode and enters privileged EXEC mode.

Step 7	**show cts interface brief**
	Example:
	`Device# show cts interface brief`
	`Interface GigabitEthernet0/0`
	`  CTS is enabled, mode:  MANUAL`
	`  Propagate SGT:  Enabled`
	`  Peer SGT assignment:  Trusted`
	`Interface GigabitEthernet0/1`
	`  CTS is enabled, mode:  MANUAL`
	`  Propagate SGT:  Disabled`
	`  Peer SGT assignment:  Untrusted`
	`Interface GigabitEthernet0/3`
	`  CTS is disabled.`
		Displays CTS configuration statistics for the interface.

Configuring CTS Credentials

SUMMARY STEPS

1. **enable**
2. **cts credentials id** *cts-id* **password** *cts-pwd*
3. **show cts credentials**
4. **show keystore**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**cts credentials id** *cts-id* **password** *cts-pwd* **Example:** `Device# cts credentials id atlas password cisco123`	Specifies the Cisco TrustSec device ID and password for this device to use when authenticating with other CTS devices with EAP-FAST.
Step 3	**show cts credentials** **Example:** `Device# show cts credentials`	Displays the Cisco TrustSec (CTS) device ID.

Step 4	**show keystore**
	Example:
	****Note that the following is the sample output of the command till Cisco IOS XE Everest release 16.5.****
	`Device# show keystore`
	`Using software keystore emulation.`
	`Keystore contains the following records (S=Simple Secret, P=PAC, R=RSA):`
	`Index Type Name`
	`----- ---- ----`
	`0 S CTS-password`
	`1 P 57366898EEF9D71A6E33C3628CE7EEDE`
	Example:
	****Note that the following is the sample output of the command from Cisco IOS XE Everest release 16.6 and above.**
	The Protected Access Credentials (PAC) information is not displayed.**
		Display the contents of the software or hardware encryption keystore.

	Command or Action	Purpose
	Device# **show keystore** Using software keystore emulation. Keystore contains the following records (S=Simple Secret, P=PAC, R=RSA): Index Type Name ----- ---- ---- 0 S CTS-password	

Example: Configuring SGT Inline Tagging

This example shows how to enable an interface on the device for L2-SGT tagging or imposition and defines whether the interface is trusted for CTS:

```
Device# configure terminal
Device(config)# interface gigabitethernet 0
Device(config-if)# cts manual
Device(config-if-cts-manual)# propagate sgt
Device(config-if-cts-manual)# policy static sgt 77 trusted
```

CHAPTER 83

Cisco TrustSec SGT Exchange Protocol IPv4

Cisco TrustSec (CTS) builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. Communication on the links between devices in the domain is secured with a combination of encryption, message integrity check, and data-path replay protection mechanisms.

The Security Group Tag (SGT) Exchange Protocol (SXP) is one of several protocols that supports CTS and is referred to in this document as CTS-SXP. CTS-SXP is a control protocol for propagating IP-to-SGT binding information across network devices that do not have the capability to tag packets. CTS-SXP passes IP to SGT bindings from authentication points to upstream devices in the network. This process allows security services on switches, routers, or firewalls to learn identity information from access devices.

Restrictions for Cisco TrustSec SGT Exchange Protocol IPv4

- The Cisco TrustSec Support for IOS feature is supported on the Cisco Integrated Services Router Generation 2 (ISR G2) only.
- CTS-SXP is supported only on physical interfaces, not on logical interfaces.
- CTS-SXP does not support IPv6.
- If the default password is configured on a router, the connection on that router should configure the password to use the default password. If the default password is not configured, the connection on that router should configure to not use the password configuration. The configuration of the password option should be consistent across the deployment network.

Information About Cisco TrustSec SGT Exchange Protocol IPv4

Security Group Tagging

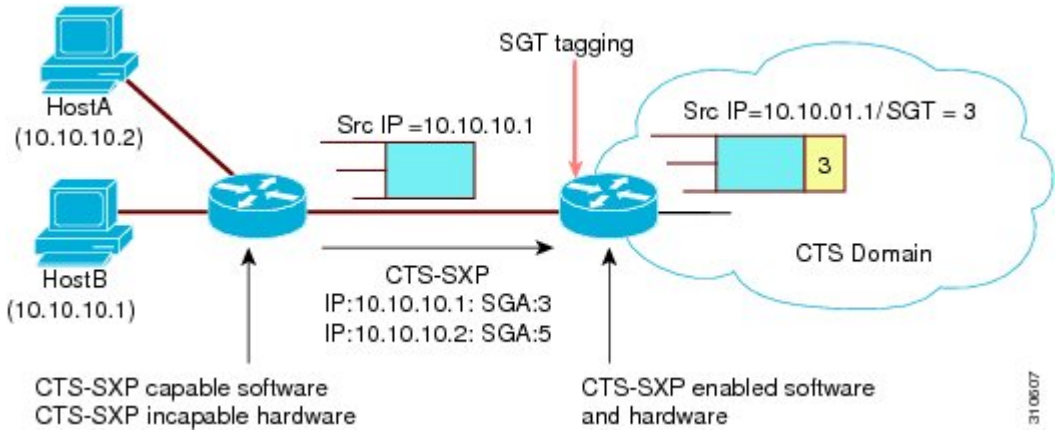
CTS-SXP uses the device and user credentials acquired during authentication for classifying the packets by security groups (SGs) as they enter the network. This packet classification is maintained by tagging packets on ingress to the CTS-SXP network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The Security Group Tag (SGT) allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Using CTS-SXP for SGT Propagation Across Legacy Access Networks

Tagging packets with SGTs requires hardware support. There may be devices in the network that can participate in CTS authentication, but lack the hardware capability to tag packets with SGTs. However, if CTS-SXP is used, then these devices can pass IP-to-SGT mappings to a CTS peer device that has CTS-capable hardware.

CTS-SXP typically operates between ingress access layer devices at the CTS domain edge and distribution layer devices within the CTS domain. The access layer device performs CTS authentication of external source devices to determine the appropriate SGTs for ingress packets. The access layer device learns the IP addresses of the source devices using IP device tracking and (optionally) DHCP snooping, then uses CTS-SXP to pass the IP addresses of the source devices along with their SGTs to the distribution switches. Distribution switches with CTS-capable hardware can use this IP-to-SGT mapping information to tag packets appropriately and to enforce Security Group Access Control List (SGACL) policies as shown in the figure below. An SGACL associates an SGT with a policy. The policy is enforced when SGT-tagged traffic egresses the CTS domain.

Figure 15: How CTS-SXP Propagates SGT Information



You must manually configure a CTS-SXP connection between a peer without CTS hardware support and a peer with CTS hardware support. The following tasks are required when configuring the CTS-SXP connection:

- If CTS-SXP data integrity and authentication are required, the same CTS-SXP password can be configured on both peer devices. The CTS-SXP password can be configured either explicitly for each peer connection or globally for the device. Although a CTS-SXP password is not required it is recommended.
- Each peer on the CTS-SXP connection must be configured as either a CTS-SXP speaker or CTS-SXP listener. The speaker device distributes the IP-to-SGT mapping information to the listener device.

- A source IP address can be specified to use for each peer relationship or a default source IP address can be configured for peer connections where a specific source IP address is not configured. If no source IP address is specified, then the device uses the interface IP address of the connection to the peer.

CTS-SXP allows multiple hops. That is, if the peer of a device lacking CTS hardware support also lacks CTS hardware support, the second peer can have a CTS-SXP connection to a third peer, continuing the propagation of the IP-to-SGT mapping information until a hardware-capable peer is reached. A device can be configured as a CTS-SXP listener for one CTS-SXP connection as a CTS-SXP speaker for another CTS-SXP connection.

A CTS device maintains connectivity with its CTS-SXP peers by using the TCP keepalive mechanism. To establish or restore a peer connection, the device repeatedly attempts the connection setup by using the configured retry period until the connection is successful or until the connection is removed from the configuration.

VRF-Aware CTS-SXP

The CTS-SXP implementation of Virtual Routing and Forwarding (VRF) binds a CTS-SXP connection with a specific VRF. It is assumed that the network topology is correctly configured for Layer 2 or Layer 3 VPNs, and that all VRFs are configured before enabling CTS-SXP.

CTS-SXP VRF support can be summarized as follows:

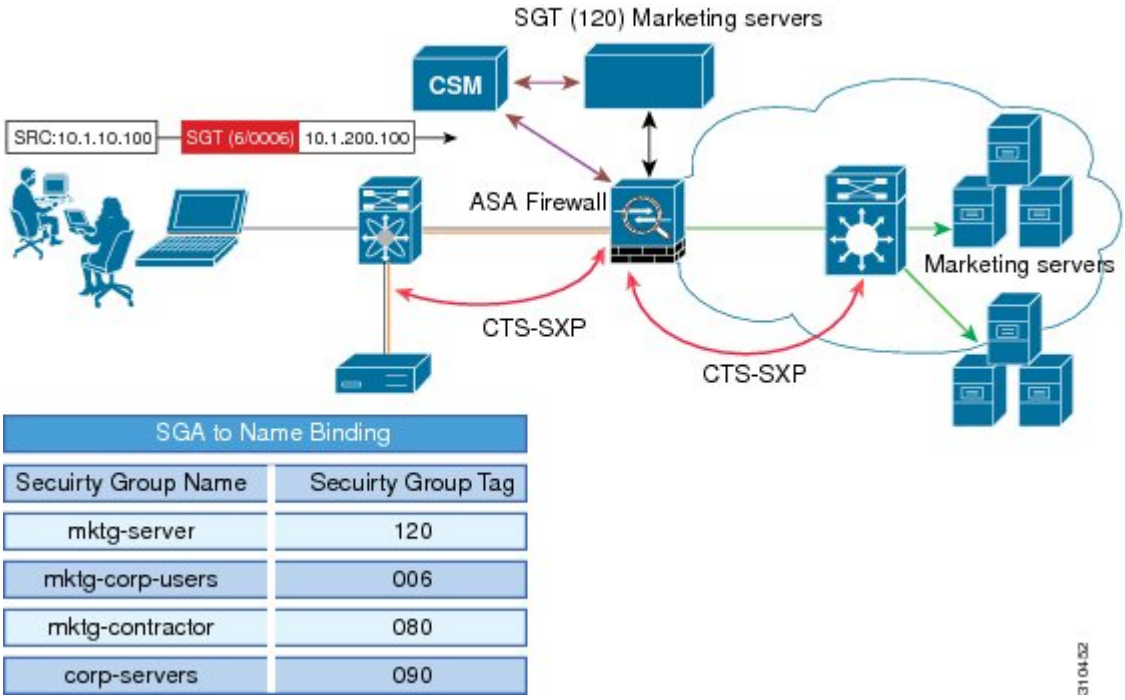
- Only one CTS-SXP connection can be bound to one VRF.
- Different VRFs may have overlapping CTS-SXP peer or source IP addresses.
- IP-to-SGT mappings learned (added or deleted) in one VRF can be updated only in the same VRF domain. The CTS-SXP connection cannot update a mapping bound to a different VRF. If no SXP connection exits for a VRF, IP-SGT mappings for that VRF will not be updated by SXP.
- CTS-SXP does not support the establishment of connections with a source IPv6 address. However, multiple address families per VRF are supported where one CTS-SXP connection in a VRF domain can forward both IPv4 and IPv6 IP-to-SGT mappings.
- CTS-SXP has no limitation on the number of connections and number of IP-to-SGT mappings per VRF.

Security Group Access Zone-Based Policy Firewall

CTS-SXP extends the deployment of network devices to additional places on the network by using the Security Group Access (SGA) Zone-Based Policy firewalls (ZBPFs). CTS-SXP is used for Identity distribution through inline devices where the identity information is learned from a primary communication path that exists across networks as shown in the figure below.

The Security Group Tag (SGT) is used by the SGA ZBPF to apply enforcement policy. IP-to-SGT mapping information is learned through CTS-SXP. When a packet arrives, source and destination IP addresses in the packet are used to derive source and destination tags. The Identity firewall applies a policy to the received IP packets based on the configured policy where the SGT is one of the attributes.

Figure 16: CTS-SXP SGA ZBPF Distribution Path Across Networks



How to Configure Cisco TrustSec SGT Exchange Protocol IPv4

Enabling CTS-SXP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cts sxp enable** **Example:** `Device(config)# cts sxp enable`	Enables a CTS-SXP connection to any peer connection that is configured. **Note** Ensure that peer connections are configured. If peer connections are not configured, then CTS-SXP connections cannot be established with them.

Configuring a CTS-SXP Peer Connection

The CTS-SXP peer connection must be configured on both devices. One device is the speaker and the other is the listener. When using password protection, make sure to use the same password on both ends.

Note If a default CTS-SXP source IP address is not configured and you do not configure a CTS-SXP source address in the connection, the Cisco TrustSec software derives the CTS-SXP source IP address from existing local IP addresses. The CTS-SXP source IP address might be different for each TCP connection initiated from the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} [[**listener** | **speaker**] [**vrf** *vrf-name*]]
4. **exit**
5. **show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp connection peer** *ipv4-address* {**source**	**password**} {**default**	**none**} **mode** {**local**	**peer**} [[**listener**	**speaker**] [**vrf** *vrf-name*]] **Example:**	Configures the CTS-SXP peer address connection. The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the

	Command or Action	Purpose
	`Device(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker`	default source address, if configured, or the address of the port. The **password** keyword specifies the password that CTS-SXP uses for the connection using the following options: • **default**—Use the default CTS-SXP password you configured using the **cts sxp default password** command. • **none**—A password is not used. The **mode** keyword specifies the role of the remote peer device: • **local**—The specified mode refers to the local device. • **peer**—The specified mode refers to the peer device. • **listener**—Specifies that the device is the listener in the connection. • **speaker**—Specifies that the device is the speaker in the connection. This is the default. The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*] **Example:** `Device# show cts sxp connections`	(Optional) Displays CTS-SXP status and connections.

Configuring the Default CTS-SXP Password

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp default password** [**0** | **6** | **7**] *password*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp default password** [**0** | **6** | **7**] *password* **Example:** `Device(config)# cts sxp default password Cisco123`	Configures the CTS-SXP default password. You can enter either a clear text password (using the **0** or no option) or an encrypted password (using the **6** or **7** option). The maximum password length is 32 characters. **Note** By default, CTS-SXP uses no password when setting up connections.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Default CTS-SXP Source IP Address

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp default source-ip** *src-ip-addr*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cts sxp default source-ip** *src-ip-addr* **Example:** `Device(config)# cts sxp default source-ip 10.20.2.2`	Configures the CTS-SXP default source IP address that is used for all new TCP connections where a source IP address is not specified. **Note** Existing TCP connections are not affected when the default CTS-SXP source IP address is configured.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the CTS-SXP Reconciliation Period

After a peer terminates a CTS-SXP connection, an internal hold-down timer starts. If the peer reconnects before the internal hold-down timer expires, the CTS-SXP reconciliation period timer starts. While the CTS-SXP reconciliation period timer is active, the CTS software retains the SGT mapping entries learned from the previous connection and removes invalid entries. The default value is 120 seconds (2 minutes). Setting the CTS-SXP reconciliation period to 0 seconds disables the timer and causes all entries from the previous connection to be removed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp reconciliation period** *seconds*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp reconciliation period** *seconds* **Example:** `Device(config)# cts sxp reconciliation period 150`	Sets the CTS-SXP reconciliation timer, in seconds. The range is from 0 to 64000. The default is 120.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring the CTS-SXP Retry Period

The CTS-SXP retry period determines how often the CTS software retries a CTS-SXP connection. If a CTS-SXP connection is not established successfully, then the CTS software makes a new attempt to set up the connection after the CTS-SXP retry period timer expires. The default value is 2 minutes. Setting the CTS-SXP retry period to 0 seconds disables the timer and retries are not attempted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp retry period** *seconds*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp retry period** *seconds* **Example:** `Device(config)# cts sxp retry period 160`	Sets the CTS-SXP retry timer, in seconds. The range is from 0 to 64000. The default is 120.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Creating Syslogs to Capture IP-to-SGT Mapping Changes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp log binding-changes**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp log binding-changes** **Example:** `Device(config)# cts sxp log binding-changes`	Enables logging for IP-to-SGT binding changes causing CTS-SXP syslogs (sev 5 syslog) to be generated whenever a change to IP-to-SGT binding occurs (add, delete, change). These changes are learned and propagated on the CTS-SXP connection. **Note** This logging function is disabled by default.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Class Map for a Security Group Access Zone-Based Policy Firewall

Perform this task to configure a class map for classifying Security Group Access (SGA) zone-based policy firewall network traffic.

Note You must perform at least one match step.

The zone-based firewall policy uses the Security Group Tag ID for filtering. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match

the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the inspect action.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group security** *name*
4. **security-group tag-id** *sgt-id*
5. **group-object** *name*
6. **description** *text*
7. **exit**
8. **class-map type inspect** [**match-any** | **match-all**] *class-map-name*
9. **match group-object security source** *name*
10. **match group-object security destination** *name*
11. **end**
12. **show object-group** [*name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**object-group security** *name* **Example:** `Device(config)# object-group security myobject1a`	Creates an object group to identify traffic coming from a specific user or endpoint and enters object-group identity mode.
Step 4	**security-group tag-id** *sgt-id* **Example:** `Device(config-object-group)# security-group tag-id 120`	Specifies the membership of a security group by using the SGT ID number. This number can be from 1 to 65535. Multiple security groups can be specified using this command.
Step 5	**group-object** *name* **Example:** `Device(config-object-group)# group-object admin`	(Optional) Specifies a nested reference to a type of user group. Multiple nested user groups can be specified using this command.

	Command or Action	Purpose
Step 6	**description** *text* **Example:** `Device(config-object-group)# description my sgtinfo`	(Optional) Defines information about the security group.
Step 7	**exit** **Example:** `Device(config-object-group)# exit`	Exits object-group identity mode and enters global configuration mode.
Step 8	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Device(config)# class-map type inspect match-any myclass1`	Creates a Layer 3 or Layer 4 inspect type class map and enters class-map configuration mode.
Step 9	**match group-object security source** *name* **Example:** `Device(config-cmap)# match group-object security source myobject1`	Matches traffic from a user in the security group.
Step 10	**match group-object security destination** *name* **Example:** `Device(config-cmap)# match group-object security destination myobject1`	Matches traffic for a user in the security group.
Step 11	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and enters privileged EXEC mode.
Step 12	**show object-group** [*name*] **Example:** `Device# show object-group admin`	(Optional) Displays the content of all user groups. Optionally, use the *name* argument to show information for a single group.

Creating a Policy Map for a Security Group Access Zone-Based Policy Firewall

Perform this task to create a policy map for a Security Group Access (SGA) zone-based policy firewall that is attached to zone pairs. This task also helps to configure Identity Firewall (IDFW) to work with Security Group Tag (SGT) Exchange Protocol (SXP) or L2-tagged traffic on the interfaces that belong to the security zones.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-name*
5. **inspect**
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **end**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **cts manual**
13. **no propagate sgt**
14. **policy static sgt** *tag* [**trusted**]
15. **exit**
16. **show policy-map type inspect zone-pair session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect z1z2-policy`	Creates a Layer 3 or Layer 4 inspect type policy map. • Enters policy map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Device(config-pmap)# class type inspect cmap-1`	Specifies the traffic (class) on which an action is to be performed and enters policy-map class configuration mode.
Step 5	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables packet inspection.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security z1z2 source z1 destination z2`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone)# service-policy type inspect z1z2-policy2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**end** **Example:** `Device(config-sec-zone)# end`	Exits security zone configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security Inside`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you should apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables the interface for Cisco TrustSec Security (CTS) SGT authorization and forwarding, and enters CTS manual interface configuration mode.
Step 13	**no propagate sgt** **Example:** `Device(config-if-cts-manual)# no propagate sgt`	Disables SGT propagation at Layer 2 on CTS interfaces.

	Command or Action	Purpose
Step 14	**policy static sgt** *tag* [**trusted**] **Example:** `Device(config-if-cts-manual)# policy static sgt 100 trusted`	Configures a static authorization policy for a CTS security group with a tagged packet that defines the trustworthiness of the SGT.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits security zone configuration mode and enters privileged EXEC mode.
Step 16	**show policy-map type inspect zone-pair session** **Example:** `Device# show policy-map type inspect zone-pair session`	(Optional) Displays the Cisco IOS stateful packet inspection sessions created because of the policy-map application on the specified zone pair. **Note** The information displayed under the class-map field is the traffic rate (bits per second) of the traffic that belongs to the connection-initiating traffic only. Unless the connection setup rate is significantly high and is sustained for multiple intervals over which the rate is computed, no significant data is shown for the connection.

Example:

The following sample output of the **show policy-map type inspect zone-pair session** command displays the information about the Cisco IOS stateful packet inspection sessions created because of the policy-map application on the specified zone pair:

```
Device# show policy-map type inspect zone-pair session

Zone-pair: in-out
 Service-policy inspect : test

   Class-map: test (match-any)
     Match: group-object security source sgt
     Inspect
       Established Sessions
        Session 113EF68C (192.2.2.1:8)=>(198.51.100.252:153) icmp SIS_OPEN
         Created 00:00:02, Last heard 00:00:02
         Bytes sent (initiator:responder) [360:360]


   Class-map: class-default (match-any)
     Match: any
     Drop (default action)
       310 packets, 37380 bytes
```

Configuration Examples for Cisco TrustSec SGT Exchange Protocol IPv4

Example: Enabling and Configuring a CTS-SXP Peer Connection

The following example shows how to enable CTS-SXP and configure the CTS-SXP peer connection on Device_A, a speaker, for connection to Device_B, a listener:

```
Device# configure terminal
Device_A(config)# cts sxp enable
Device_A(config)# cts sxp default password Cisco123
Device_A(config)# cts sxp default source-ip 10.10.1.1
Device_A(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker
```

The following example shows how to configure the CTS-SXP peer connection on Device_B, a listener, for connection to Device_A, a speaker:

```
Device# configure terminal
Device_B(config)# cts sxp enable
Device_B(config)# cts sxp default password Cisco123
Device_B(config)# cts sxp default source-ip 10.20.2.2
Device_B(config)# cts sxp connection peer 10.10.1.1 password default mode local listener
```

The following sample output for **show cts sxp connections** command displays CTS-SXP connections:

```
Device_B# show cts sxp connections

 SXP              : Enabled
 Default Password : Set
 Default Source IP: 10.10.1.1
Connection retry open period: 10 secs
Reconcile period: 120 secs
Retry open timer is not running
----------------------------------------------
Peer IP          : 10.20.2.2
Source IP        : 10.10.1.1
Conn status      : On
Connection mode  : SXP Listener
Connection inst# : 1
TCP conn fd      : 1
TCP conn password: default SXP password
Duration since last state change: 0:00:21:25 (dd:hr:mm:sec)
Total num of SXP Connections = 1
```

Example: Configuring a Security Group Access Zone-Based Policy Firewall

The following example shows the configuration of a class map and policy map for an SGA zone-based policy firewall.

```
Device(config)# object-group security myobject1
Device(config-object-group)# security-group tag-id 1
Device(config-object-group)# exit
Device(config)# object-group security myobject2
```

```
Device(config-object-group)# security-group tag-id 2
Device(config-object-group)# exit
Device(config)# object-group security myobject3
Device(config-object-group)# security-group tag-id 3
Device(config-object-group)# exit
Device(config)# object-group security myobject4
Device(config-object-group)# security-group tag-id 4
Device(config-object-group)# exit

Device(config)# class-map type inspect match-any myclass1
Device(config-cmap)# match group-object security source myobject1
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass2
Device(config-cmap)# match group-object security source myobject2
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass3
Device(config-cmap)# match group-object security source myobject3
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass4
Device(config-cmap)# match group-object security source myobject4
Device(config-cmap)# exit

Device(config)# policy-map type inspect InsideOutside
Device(config-pmap)# class type inspect myclass1
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class type inspect myclass2
Device(config-pmap-c)# drop log
Device(config-pmap-c)# exit

Device(config)# policy-map type inspect OutsideInside
Device(config-pmap)# class type inspect myclass3
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class type inspect myclass4
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit

Device(config)# zone-pair security Inside
Device(config-sec-zone)# description Firewall Inside Zone
Device(config-sec-zone)# exit

Device(config)# zone-pair security Outside
Device(config-sec-zone)# description Firewall Outside Zone
Device(config-sec-zone)# exit

Device(config)# zone-pair security InsideOutside source Inside destination Outside
Device(config-sec-zone)# description Firewall ZonePair Inside Outside
Device(config-sec-zone)# service-policy type inspect InsideOutside
Device(config-sec-zone)# exit

Device(config)# zone-pair security OutsideInside source Outside destination Inside
Device(config-sec-zone)# description Firewall ZonePair Outside Inside
Device(config-sec-zone)# service-policy type inspect OutsideInside
Device(config-sec-zone)# exit

Device(config)# interface Gigabit 0/1/1
Device(config-if)# zone-member security Inside
Device(config-if)# exit
```

Additional References for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference: Commands A to C
	Cisco IOS Security Command Reference: Commands D to L
	Cisco IOS Security Command Reference: Commands M to R
	Cisco IOS Security Command Reference: Commands S to Z
Cisco TrustSec switches	Cisco TrustSec Switch Configuration Guide

MIBs

MIB	MIBs Link
CISCO-TRUSTSEC-SXP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

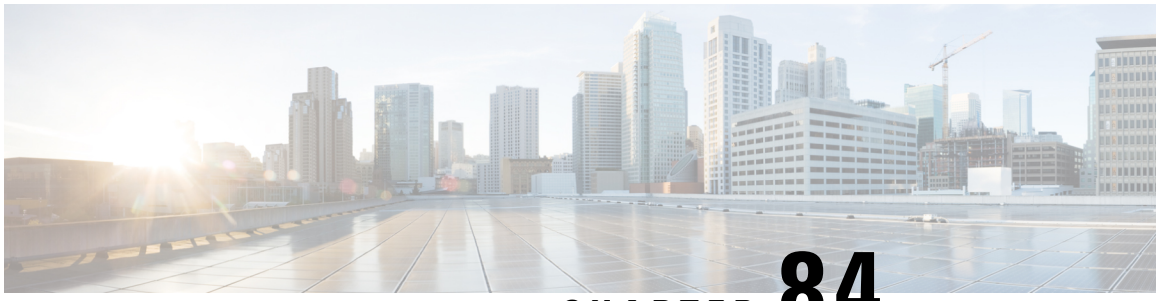
Feature Information for Cisco TrustSec SGT Exchange Protocol IPv4

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 118: Feature Information for Cisco TrustSec SGT Exchange Protocol IPv4

Feature Name	Releases	Feature Information
Cisco TrustSec SGT Exchange Protocol IPv4		The Security Group Tag (SGT) Exchange Protocol (SXP) is one of several protocols that supports CTS and is referred to in this document as CTS-SXP. CTS-SXP is a control protocol for propagating IP-to-SGT binding information across network devices that do not have the capability to tag packets. CTS-SXP passes IP-to-SGT bindings from authentication points to upstream devices in the network. This allows security services on switches, routers, or firewalls to learn identity information from access devices. The following commands were introduced or modified: **cts sxp enable, cts sxp connection peer**, **show cts sxp**, **cts sxp default source-ip, cts sxp reconciliation period, cts sxp retry period, cts sxp log binding-changes**.
TrustSec SG Firewall Enforcement IPv4		This feature helps CTS-SXP extend the deployment of network devices through Security Group Access (SGA) Zone-Based Policy firewalls (ZBPFs). The following commands were introduced or modified: **group-object**, **match group-object security**, **object-group security**, **policy static sgt**, and **security-group**.



CHAPTER 84

TrustSec SGT Handling: L2 SGT Imposition and Forwarding

First Published: July 25, 2011

Cisco TrustSec (CTS) builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. Communication on the links between devices in the domain is secured with a combination of encryption, message integrity check, and data-path replay protection mechanisms.

The TrustSec SGT Handling: L2 SGT Imposition and Forwarding feature allows the interfaces in a router to be manually enabled for CTS so that the router can insert the Security Group Tag (SGT) in the packet to be carried throughout the network in the CTS header.

Prerequisites for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

The CTS network needs to be established with the following prerequisites before implementing the TrustSec SGT Handling: L2 SGT Imposition and Forwarding feature:

- Connectivity exists between all network devices
- Cisco Secure Access Control System (ACS) 5.1 operates with a CTS-SXP license
- Directory, DHCP, DNS, certificate authority, and NTP servers function within the network
- Configure the **retry open timer** command to a different value on different routers.

Information about TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Security Groups and SGTs

A security group is a grouping of users, endpoint devices, and resources that share access control policies. Security groups are defined by the administrator in the ACS. As new users and devices are added to the Cisco TrustSec (CTS) domain, the authentication server assigns these new entities to appropriate security groups. CTS assigns to each security group a unique 16-bit security group number whose scope is global within a CTS domain. The number of security groups in the router is limited to the number of authenticated network entities. Security group numbers do not need to be manually configured.

Once a device is authenticated, CTS tags any packet that originates from that device with an SGT that contains the security group number of the device. The packet carries this SGT throughout the network within the CTS header. The SGT is a single label that determines the privileges of the source within the entire CTS domain. The SGT is identified as the source because it contains the security group of the source. The destination device is assigned a destination group tag (DGT).

Note The CTS packet tag does not contain the security group number of the destination device.

How to Configure TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Manually Enabling TrustSec SGT Handling: L2 SGT Imposition and Forwarding on an Interface

Perform the following steps to manually enable an interface on the device for Cisco TrustSec (CTS) so that the device can add Security Group Tag (SGT) in the packet to be propagated throughout the network and to implement a static authorization policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**GigabitEthernet** *port* | **Vlan** *number*}
4. **cts manual**
5. **policy static sgt** *tag* [trusted]
6. **end**
7. **show cts interface** [**GigabitEthernet** *port* | **Vlan** *number* | **brief** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**GigabitEthernet** *port* | **Vlan** *number*} **Example:** `Device(config)# interface gigabitethernet 0`	Enters the interface on which CTS SGT authorization and forwarding is enabled
Step 4	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables the interface for CTS SGT authorization and forwarding, and enters CTS manual interface configuration mode. **Note** To enable the **cts manual** command on a subinterface, you must increase the IP MTU size to accommodate the additional bytes for the Dot1Q tag. This is applicable only for releases earlier than Cisco IOS XE Release 3.17.
Step 5	**policy static sgt** *tag* [trusted] **Example:** `Device(config-if-cts-manual)# policy static sgt 100 trusted`	Configures a static authorization policy for a CTS security group with a tagged packet that defines the trustworthiness of the SGT.
Step 6	**end** **Example:** `Device(config-if-cts-manual)# end`	Exits CTS manual interface configuration mode and enters privileged EXEC mode.
Step 7	**show cts interface** [**GigabitEthernet** *port* | **Vlan** *number* | **brief** | **summary**] **Example:** `Device# show cts interface brief`	Displays CTS configuration statistics for the interface.

Example:

The following is sample output for the **show cts interface brief**command.

Cisco ASR 1000 Series Aggregation Services Routers and Cisco Cloud Services Router 1000V Series

```
Device# show cts interface brief
```

```
Global Dot1x feature is Disabled
 Interface GigabitEthernet0/1/0:
     CTS is enabled, mode:    MANUAL
     IFC state:               OPEN
     Interface Active for 00:00:40.386
     Authentication Status:   NOT APPLICABLE
         Peer identity:       "unknown"
         Peer's advertised capabilities: ""
     Authorization Status:    NOT APPLICABLE
     SAP Status:              NOT APPLICABLE
     Propagate SGT:           Enabled
     Cache Info:
         Cache applied to link : NONE
```

Cisco 4400 Series Integrated Services Routers

```
Device# show cts interface brief

Interface GigabitEthernet0/1/0
    CTS is enabled, mode:     MANUAL
      Propagate SGT:          Enabled
      Static Ingress SGT Policy:
        Peer SGT:             100
        Peer SGT assignment:  Trusted
```

Disabling CTS SGT Propagation on an Interface

Follow these steps to disable CTS SGT Propagation on an interface in an instance when a peer device is not capable of receiving an SGT.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**GigabitEthernet***port* | **Vlan** *number*}
4. **cts manual**
5. **no propagate sgt**
6. **end**
7. **show cts interface** [**GigabitEthernet***port* | **Vlan** *number* | **brief** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** {**GigabitEthernet***port* | **Vlan** *number*} **Example:** `Device(config)# interface gigabitethernet 0`	Enters the interface on which CTS SGT authorization and forwarding is enabled
Step 4	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables the interface for CTS SGT authorization and forwarding. CTS manual interface configuration mode is entered where CTS parameters can be configured.
Step 5	**no propagate sgt** **Example:** `Device(config-if-cts-manual)# no propagate sgt`	Disables CTS SGT propagation on an interface in situations where a peer device is not capable of receiving an SGT. **Note** CTS SGT propagation is enabled by default. The **propagate sgt** command can be used if CTS SGT propagation needs to be turned on again for a peer device. Once the **no propagate sgt** command is entered, the SGT tag is not added in the L2 header.
Step 6	**end** **Example:** `Device(config-if-cts-manual)# end`	Exits CTS manual interface configuration mode and enters privileged EXEC mode.

Step 7	**show cts interface** [**GigabitEthernet***port* | **Vlan** *number* | **brief** | **summary**]
	Example:
	`Device# show cts interface brief`
	`Global Dot1x feature is Disabled`
	`Interface GigabitEthernet0:`
	`CTS is enabled, mode: MANUAL`
	`IFC state: OPEN`
	`Authentication Status: NOT APPLICABLE`
	`Peer identity: "unknown"`
	`Peer's advertised capabilities: ""`
	`Authorization Status: NOT APPLICABLE`
	`SAP Status: NOT APPLICABLE`
	`Propagate SGT: Disabled`
	`Cache Info:`
	`Cache applied to link : NONE`
		Displays CTS configuration statistics to verify that CTS SGT propagation was disabled on interface.

Additional References for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference: Commands A to C
	Cisco IOS Security Command Reference: Commands D to L
	Cisco IOS Security Command Reference: Commands M to R
	Cisco IOS Security Command Reference: Commands S to Z
Cisco TrustSec switches	Cisco TrustSec Switch Configuration Guide

MIBs

MIB	MIBs Link
CISCO-TRUSTSEC-SXP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 119: Feature Information for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Feature Name	Releases	Feature Information
TrustSec SGT Handling: L2 SGT Imposition and Forwarding		This feature allows the interfaces in a router to be manually enabled for CTS so that the router can insert the Security Group Tag (SGT) in the packet to be carried throughout the network in the CTS header. • Cisco CSR 1000V Router • Cisco ISR 4400 Router • Catalyst 3850 Series Switches • Catalyst 3650 Series Switches • Cisco 5700 Series Wireless LAN Controllers • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E • Cisco Catalyst 4500-X Series Switches • Cisco Catalyst 4500E Supervisor Engine 8-E • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3650 Series Switches The following commands were introduced or modified: **cts manual**, **policy static sgt**, **propagate sgt**, **show cts interface**.

CHAPTER 85

Prerequisites for Cisco TrustSec SGT Exchange Protocol IPv4

The CTS-SXP network needs to be established before implementing SXP. The CTS-SXP network has the following prerequisites:

- To use the Cisco TrustSec functionality on your existing router, ensure that you have purchased a Cisco TrustSec security license. If the router is being ordered and needs the Cisco TrustSec functionality, ensure that this license is pre-installed on your router before it is shipped to you.
- CTS-SXP software runs on all network devices
- Connectivity exists between all network devices
- The Cisco Identity Services Engine 1.0 is required for authentication. The Secure Access Control Server (ACS) Express Appliance server can also be used for authentication, however not all ACS features are supported by CTS. ACS 5.1 operates with a CTS-SXP license.
- Configure the **retry open timer** command to a different value on different routers.

Restrictions for Cisco TrustSec SGT Exchange Protocol IPv4

- The Cisco TrustSec Support for IOS feature is supported on the Cisco Integrated Services Router Generation 2 (ISR G2) only.
- CTS-SXP is supported only on physical interfaces, not on logical interfaces.
- CTS-SXP does not support IPv6.
- If the default password is configured on a router, the connection on that router should configure the password to use the default password. If the default password is not configured, the connection on that

router should configure to not use the password configuration. The configuration of the password option should be consistent across the deployment network.

Information About Cisco TrustSec SGT Exchange Protocol IPv4

Security Group Tagging

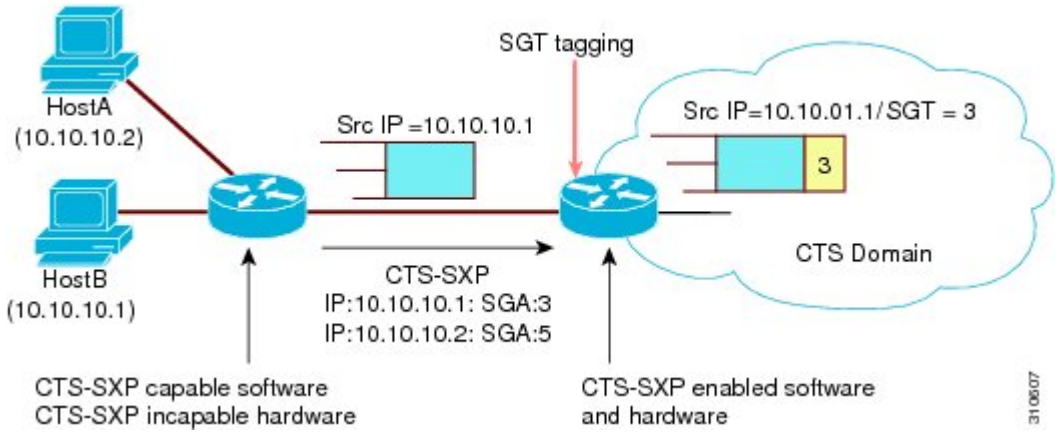
CTS-SXP uses the device and user credentials acquired during authentication for classifying the packets by security groups (SGs) as they enter the network. This packet classification is maintained by tagging packets on ingress to the CTS-SXP network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The Security Group Tag (SGT) allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Using CTS-SXP for SGT Propagation Across Legacy Access Networks

Tagging packets with SGTs requires hardware support. There may be devices in the network that can participate in CTS authentication, but lack the hardware capability to tag packets with SGTs. However, if CTS-SXP is used, then these devices can pass IP-to-SGT mappings to a CTS peer device that has CTS-capable hardware.

CTS-SXP typically operates between ingress access layer devices at the CTS domain edge and distribution layer devices within the CTS domain. The access layer device performs CTS authentication of external source devices to determine the appropriate SGTs for ingress packets. The access layer device learns the IP addresses of the source devices using IP device tracking and (optionally) DHCP snooping, then uses CTS-SXP to pass the IP addresses of the source devices along with their SGTs to the distribution switches. Distribution switches with CTS-capable hardware can use this IP-to-SGT mapping information to tag packets appropriately and to enforce Security Group Access Control List (SGACL) policies as shown in the figure below. An SGACL associates an SGT with a policy. The policy is enforced when SGT-tagged traffic egresses the CTS domain.

Figure 17: How CTS-SXP Propagates SGT Information



You must manually configure a CTS-SXP connection between a peer without CTS hardware support and a peer with CTS hardware support. The following tasks are required when configuring the CTS-SXP connection:

- If CTS-SXP data integrity and authentication are required, the same CTS-SXP password can be configured on both peer devices. The CTS-SXP password can be configured either explicitly for each peer connection or globally for the device. Although a CTS-SXP password is not required it is recommended.
- Each peer on the CTS-SXP connection must be configured as either a CTS-SXP speaker or CTS-SXP listener. The speaker device distributes the IP-to-SGT mapping information to the listener device.
- A source IP address can be specified to use for each peer relationship or a default source IP address can be configured for peer connections where a specific source IP address is not configured. If no source IP address is specified, then the device uses the interface IP address of the connection to the peer.

CTS-SXP allows multiple hops. That is, if the peer of a device lacking CTS hardware support also lacks CTS hardware support, the second peer can have a CTS-SXP connection to a third peer, continuing the propagation of the IP-to-SGT mapping information until a hardware-capable peer is reached. A device can be configured as a CTS-SXP listener for one CTS-SXP connection as a CTS-SXP speaker for another CTS-SXP connection.

A CTS device maintains connectivity with its CTS-SXP peers by using the TCP keepalive mechanism. To establish or restore a peer connection, the device repeatedly attempts the connection setup by using the configured retry period until the connection is successful or until the connection is removed from the configuration.

VRF-Aware CTS-SXP

The CTS-SXP implementation of Virtual Routing and Forwarding (VRF) binds a CTS-SXP connection with a specific VRF. It is assumed that the network topology is correctly configured for Layer 2 or Layer 3 VPNs, and that all VRFs are configured before enabling CTS-SXP.

CTS-SXP VRF support can be summarized as follows:

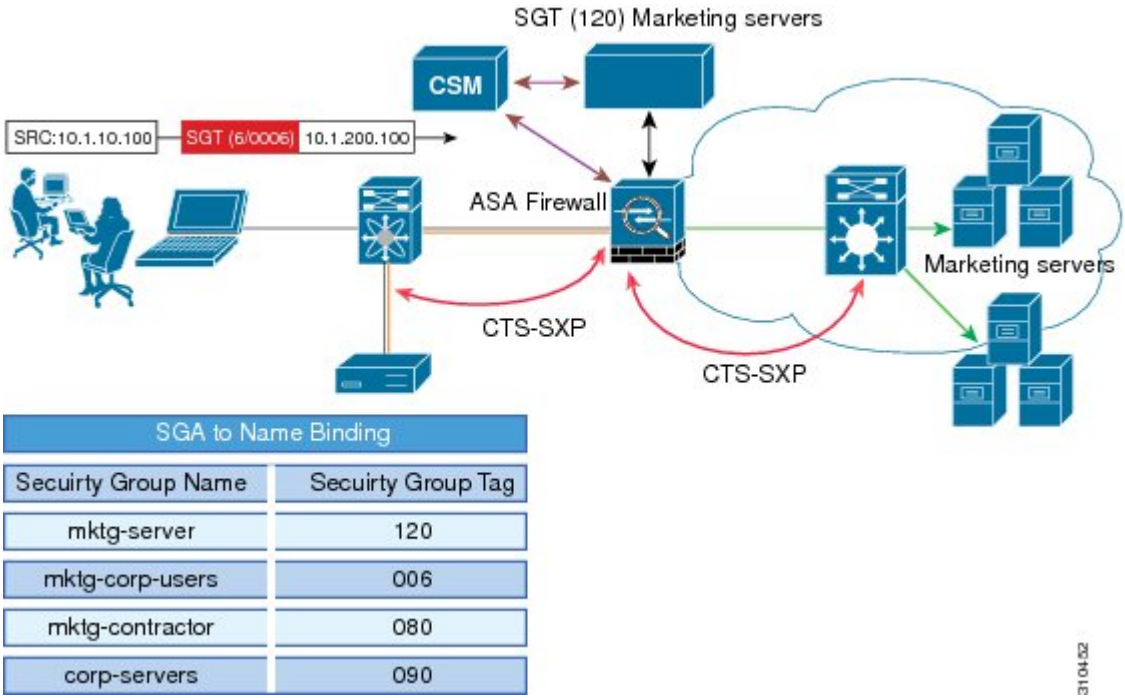
- Only one CTS-SXP connection can be bound to one VRF.
- Different VRFs may have overlapping CTS-SXP peer or source IP addresses.
- IP-to-SGT mappings learned (added or deleted) in one VRF can be updated only in the same VRF domain. The CTS-SXP connection cannot update a mapping bound to a different VRF. If no SXP connection exits for a VRF, IP-SGT mappings for that VRF will not be updated by SXP.
- CTS-SXP does not support the establishment of connections with a source IPv6 address. However, multiple address families per VRF are supported where one CTS-SXP connection in a VRF domain can forward both IPv4 and IPv6 IP-to-SGT mappings.
- CTS-SXP has no limitation on the number of connections and number of IP-to-SGT mappings per VRF.

Security Group Access Zone-Based Policy Firewall

CTS-SXP extends the deployment of network devices to additional places on the network by using the Security Group Access (SGA) Zone-Based Policy firewalls (ZBPFs). CTS-SXP is used for Identity distribution through inline devices where the identity information is learned from a primary communication path that exists across networks as shown in the figure below.

The Security Group Tag (SGT) is used by the SGA ZBPF to apply enforcement policy. IP-to-SGT mapping information is learned through CTS-SXP. When a packet arrives, source and destination IP addresses in the packet are used to derive source and destination tags. The Identity firewall applies a policy to the received IP packets based on the configured policy where the SGT is one of the attributes.

Figure 18: CTS-SXP SGA ZBPF Distribution Path Across Networks



SGA to Name Binding	
Secuirty Group Name	Secuirty Group Tag
mktg-server	120
mktg-corp-users	006
mktg-contractor	080
corp-servers	090

How to Configure Cisco TrustSec SGT Exchange Protocol IPv4

Enabling CTS-SXP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cts sxp enable** **Example:** `Device(config)# cts sxp enable`	Enables a CTS-SXP connection to any peer connection that is configured. **Note** Ensure that peer connections are configured. If peer connections are not configured, then CTS-SXP connections cannot be established with them.

Configuring a CTS-SXP Peer Connection

The CTS-SXP peer connection must be configured on both devices. One device is the speaker and the other is the listener. When using password protection, make sure to use the same password on both ends.

Note If a default CTS-SXP source IP address is not configured and you do not configure a CTS-SXP source address in the connection, the Cisco TrustSec software derives the CTS-SXP source IP address from existing local IP addresses. The CTS-SXP source IP address might be different for each TCP connection initiated from the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} [[**listener** | **speaker**] [**vrf** *vrf-name*]]
4. **exit**
5. **show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp connection peer** *ipv4-address* {**source**	**password**} {**default**	**none**} **mode** {**local**	**peer**} [[**listener**	**speaker**] [**vrf** *vrf-name*]] **Example:**	Configures the CTS-SXP peer address connection. The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the

	Command or Action	Purpose
	`Device(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker`	default source address, if configured, or the address of the port. The **password** keyword specifies the password that CTS-SXP uses for the connection using the following options: • **default**—Use the default CTS-SXP password you configured using the **cts sxp default password** command. • **none**—A password is not used. The **mode** keyword specifies the role of the remote peer device: • **local**—The specified mode refers to the local device. • **peer**—The specified mode refers to the peer device. • **listener**—Specifies that the device is the listener in the connection. • **speaker**—Specifies that the device is the speaker in the connection. This is the default. The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*] **Example:** `Device# show cts sxp connections`	(Optional) Displays CTS-SXP status and connections.

Configuring the Default CTS-SXP Password

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp default password** [**0** | **6** | **7**] *password*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp default password** [**0** | **6** | **7**] *password* **Example:** `Device(config)# cts sxp default password Cisco123`	Configures the CTS-SXP default password. You can enter either a clear text password (using the **0** or no option) or an encrypted password (using the **6** or **7** option). The maximum password length is 32 characters. **Note** By default, CTS-SXP uses no password when setting up connections.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Default CTS-SXP Source IP Address

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp default source-ip** *src-ip-addr*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cts sxp default source-ip** *src-ip-addr* **Example:** `Device(config)# cts sxp default source-ip 10.20.2.2`	Configures the CTS-SXP default source IP address that is used for all new TCP connections where a source IP address is not specified. **Note** Existing TCP connections are not affected when the default CTS-SXP source IP address is configured.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the CTS-SXP Reconciliation Period

After a peer terminates a CTS-SXP connection, an internal hold-down timer starts. If the peer reconnects before the internal hold-down timer expires, the CTS-SXP reconciliation period timer starts. While the CTS-SXP reconciliation period timer is active, the CTS software retains the SGT mapping entries learned from the previous connection and removes invalid entries. The default value is 120 seconds (2 minutes). Setting the CTS-SXP reconciliation period to 0 seconds disables the timer and causes all entries from the previous connection to be removed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp reconciliation period** *seconds*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp reconciliation period** *seconds* **Example:** `Device(config)# cts sxp reconciliation period 150`	Sets the CTS-SXP reconciliation timer, in seconds. The range is from 0 to 64000. The default is 120.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring the CTS-SXP Retry Period

The CTS-SXP retry period determines how often the CTS software retries a CTS-SXP connection. If a CTS-SXP connection is not established successfully, then the CTS software makes a new attempt to set up the connection after the CTS-SXP retry period timer expires. The default value is 2 minutes. Setting the CTS-SXP retry period to 0 seconds disables the timer and retries are not attempted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp retry period** *seconds*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp retry period** *seconds* **Example:** `Device(config)# cts sxp retry period 160`	Sets the CTS-SXP retry timer, in seconds. The range is from 0 to 64000. The default is 120.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Creating Syslogs to Capture IP-to-SGT Mapping Changes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp log binding-changes**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp log binding-changes** **Example:** `Device(config)# cts sxp log binding-changes`	Enables logging for IP-to-SGT binding changes causing CTS-SXP syslogs (sev 5 syslog) to be generated whenever a change to IP-to-SGT binding occurs (add, delete, change). These changes are learned and propagated on the CTS-SXP connection. **Note** This logging function is disabled by default.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Class Map for a Security Group Access Zone-Based Policy Firewall

Perform this task to configure a class map for classifying Security Group Access (SGA) zone-based policy firewall network traffic.

Note You must perform at least one match step.

The zone-based firewall policy uses the Security Group Tag ID for filtering. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match

the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the inspect action.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group security** *name*
4. **security-group tag-id** *sgt-id*
5. **group-object** *name*
6. **description** *text*
7. **exit**
8. **class-map type inspect** [**match-any** | **match-all**] *class-map-name*
9. **match group-object security source** *name*
10. **match group-object security destination** *name*
11. **end**
12. **show object-group** [*name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**object-group security** *name* **Example:** `Device(config)# object-group security myobject1a`	Creates an object group to identify traffic coming from a specific user or endpoint and enters object-group identity mode.
Step 4	**security-group tag-id** *sgt-id* **Example:** `Device(config-object-group)# security-group tag-id 120`	Specifies the membership of a security group by using the SGT ID number. This number can be from 1 to 65535. Multiple security groups can be specified using this command.
Step 5	**group-object** *name* **Example:** `Device(config-object-group)# group-object admin`	(Optional) Specifies a nested reference to a type of user group. Multiple nested user groups can be specified using this command.

	Command or Action	Purpose
Step 6	**description** *text* **Example:** `Device(config-object-group)# description my sgtinfo`	(Optional) Defines information about the security group.
Step 7	**exit** **Example:** `Device(config-object-group)# exit`	Exits object-group identity mode and enters global configuration mode.
Step 8	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Device(config)# class-map type inspect match-any myclass1`	Creates a Layer 3 or Layer 4 inspect type class map and enters class-map configuration mode.
Step 9	**match group-object security source** *name* **Example:** `Device(config-cmap)# match group-object security source myobject1`	Matches traffic from a user in the security group.
Step 10	**match group-object security destination** *name* **Example:** `Device(config-cmap)# match group-object security destination myobject1`	Matches traffic for a user in the security group.
Step 11	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and enters privileged EXEC mode.
Step 12	**show object-group** [*name*] **Example:** `Device# show object-group admin`	(Optional) Displays the content of all user groups. Optionally, use the *name* argument to show information for a single group.

Creating a Policy Map for a Security Group Access Zone-Based Policy Firewall

Perform this task to create a policy map for a Security Group Access (SGA) zone-based policy firewall that is attached to zone pairs. This task also helps to configure Identity Firewall (IDFW) to work with Security Group Tag (SGT) Exchange Protocol (SXP) or L2-tagged traffic on the interfaces that belong to the security zones.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-name*
5. **inspect**
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **end**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **cts manual**
13. **no propagate sgt**
14. **policy static sgt** *tag* [**trusted**]
15. **exit**
16. **show policy-map type inspect zone-pair session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect z1z2-policy`	Creates a Layer 3 or Layer 4 inspect type policy map. • Enters policy map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Device(config-pmap)# class type inspect cmap-1`	Specifies the traffic (class) on which an action is to be performed and enters policy-map class configuration mode.
Step 5	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables packet inspection.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security z1z2 source z1 destination z2`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone)# service-policy type inspect z1z2-policy2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**end** **Example:** `Device(config-sec-zone)# end`	Exits security zone configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security Inside`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you should apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables the interface for Cisco TrustSec Security (CTS) SGT authorization and forwarding, and enters CTS manual interface configuration mode.
Step 13	**no propagate sgt** **Example:** `Device(config-if-cts-manual)# no propagate sgt`	Disables SGT propagation at Layer 2 on CTS interfaces.

	Command or Action	Purpose
Step 14	**policy static sgt** *tag* [**trusted**] **Example:** `Device(config-if-cts-manual)# policy static sgt 100 trusted`	Configures a static authorization policy for a CTS security group with a tagged packet that defines the trustworthiness of the SGT.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits security zone configuration mode and enters privileged EXEC mode.
Step 16	**show policy-map type inspect zone-pair session** **Example:** `Device# show policy-map type inspect zone-pair session`	(Optional) Displays the Cisco IOS stateful packet inspection sessions created because of the policy-map application on the specified zone pair. **Note** The information displayed under the class-map field is the traffic rate (bits per second) of the traffic that belongs to the connection-initiating traffic only. Unless the connection setup rate is significantly high and is sustained for multiple intervals over which the rate is computed, no significant data is shown for the connection.

Example:

The following sample output of the **show policy-map type inspect zone-pair session** command displays the information about the Cisco IOS stateful packet inspection sessions created because of the policy-map application on the specified zone pair:

```
Device# show policy-map type inspect zone-pair session

Zone-pair: in-out
 Service-policy inspect : test

   Class-map: test (match-any)
     Match: group-object security source sgt
     Inspect
       Established Sessions
        Session 113EF68C (192.2.2.1:8)=>(198.51.100.252:153) icmp SIS_OPEN
         Created 00:00:02, Last heard 00:00:02
         Bytes sent (initiator:responder) [360:360]


   Class-map: class-default (match-any)
     Match: any
     Drop (default action)
       310 packets, 37380 bytes
```

Configuration Examples for Cisco TrustSec SGT Exchange Protocol IPv4

Example: Enabling and Configuring a CTS-SXP Peer Connection

The following example shows how to enable CTS-SXP and configure the CTS-SXP peer connection on Device_A, a speaker, for connection to Device_B, a listener:

```
Device# configure terminal
Device_A(config)# cts sxp enable
Device_A(config)# cts sxp default password Cisco123
Device_A(config)# cts sxp default source-ip 10.10.1.1
Device_A(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker
```

The following example shows how to configure the CTS-SXP peer connection on Device_B, a listener, for connection to Device_A, a speaker:

```
Device# configure terminal
Device_B(config)# cts sxp enable
Device_B(config)# cts sxp default password Cisco123
Device_B(config)# cts sxp default source-ip 10.20.2.2
Device_B(config)# cts sxp connection peer 10.10.1.1 password default mode local listener
```

The following sample output for **show cts sxp connections** command displays CTS-SXP connections:

```
Device_B# show cts sxp connections

 SXP              : Enabled
 Default Password : Set
 Default Source IP: 10.10.1.1
Connection retry open period: 10 secs
Reconcile period: 120 secs
Retry open timer is not running
------------------------------------------------
Peer IP          : 10.20.2.2
Source IP        : 10.10.1.1
Conn status      : On
Connection mode  : SXP Listener
Connection inst# : 1
TCP conn fd      : 1
TCP conn password: default SXP password
Duration since last state change: 0:00:21:25 (dd:hr:mm:sec)
Total num of SXP Connections = 1
```

Example: Configuring a Security Group Access Zone-Based Policy Firewall

The following example shows the configuration of a class map and policy map for an SGA zone-based policy firewall.

```
Device(config)# object-group security myobject1
Device(config-object-group)# security-group tag-id 1
Device(config-object-group)# exit
Device(config)# object-group security myobject2
```

```
Device(config-object-group)# security-group tag-id 2
Device(config-object-group)# exit
Device(config)# object-group security myobject3
Device(config-object-group)# security-group tag-id 3
Device(config-object-group)# exit
Device(config)# object-group security myobject4
Device(config-object-group)# security-group tag-id 4
Device(config-object-group)# exit

Device(config)# class-map type inspect match-any myclass1
Device(config-cmap)# match group-object security source myobject1
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass2
Device(config-cmap)# match group-object security source myobject2
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass3
Device(config-cmap)# match group-object security source myobject3
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any myclass4
Device(config-cmap)# match group-object security source myobject4
Device(config-cmap)# exit

Device(config)# policy-map type inspect InsideOutside
Device(config-pmap)# class type inspect myclass1
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class type inspect myclass2
Device(config-pmap-c)# drop log
Device(config-pmap-c)# exit

Device(config)# policy-map type inspect OutsideInside
Device(config-pmap)# class type inspect myclass3
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class type inspect myclass4
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit

Device(config)# zone-pair security Inside
Device(config-sec-zone)# description Firewall Inside Zone
Device(config-sec-zone)# exit

Device(config)# zone-pair security Outside
Device(config-sec-zone)# description Firewall Outside Zone
Device(config-sec-zone)# exit

Device(config)# zone-pair security InsideOutside source Inside destination Outside
Device(config-sec-zone)# description Firewall ZonePair Inside Outside
Device(config-sec-zone)# service-policy type inspect InsideOutside
Device(config-sec-zone)# exit

Device(config)# zone-pair security OutsideInside source Outside destination Inside
Device(config-sec-zone)# description Firewall ZonePair Outside Inside
Device(config-sec-zone)# service-policy type inspect OutsideInside
Device(config-sec-zone)# exit

Device(config)# interface Gigabit 0/1/1
Device(config-if)# zone-member security Inside
Device(config-if)# exit
```

Additional References for TrustSec SGT Handling: L2 SGT Imposition and Forwarding

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference: Commands A to C
	Cisco IOS Security Command Reference: Commands D to L
	Cisco IOS Security Command Reference: Commands M to R
	Cisco IOS Security Command Reference: Commands S to Z
Cisco TrustSec switches	Cisco TrustSec Switch Configuration Guide

MIBs

MIB	MIBs Link
CISCO-TRUSTSEC-SXP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

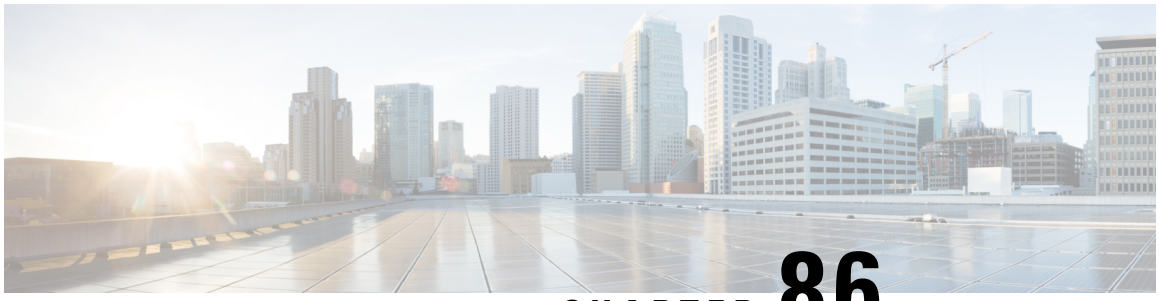
Feature Information for Cisco TrustSec SGT Exchange Protocol IPv4

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 120: Feature Information for Cisco TrustSec SGT Exchange Protocol IPv4

Feature Name	Releases	Feature Information
Cisco TrustSec SGT Exchange Protocol IPv4		The Security Group Tag (SGT) Exchange Protocol (SXP) is one of several protocols that supports CTS and is referred to in this document as CTS-SXP. CTS-SXP is a control protocol for propagating IP-to-SGT binding information across network devices that do not have the capability to tag packets. CTS-SXP passes IP-to-SGT bindings from authentication points to upstream devices in the network. This allows security services on switches, routers, or firewalls to learn identity information from access devices. The following commands were introduced or modified: **cts sxp enable, cts sxp connection peer**, **show cts sxp**, **cts sxp default source-ip, cts sxp reconciliation period, cts sxp retry period, cts sxp log binding-changes**.
TrustSec SG Firewall Enforcement IPv4		This feature helps CTS-SXP extend the deployment of network devices through Security Group Access (SGA) Zone-Based Policy firewalls (ZBPFs). The following commands were introduced or modified: **group-object**, **match group-object security**, **object-group security**, **policy static sgt**, and **security-group**.



CHAPTER 86

Enabling Bidirectional SXP Support

The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection.

Prerequisites for Bidirectional SXP Support

- Ensure that Cisco TrustSec is configured on the device. For more information, see the "Cisco TrustSec Support for IOS" chapter in the *Cisco TrustSec Configuration Guide*.
- To use the Cisco TrustSec functionality on your existing device, ensure that you have purchased one of the following security licenses:
  - IP Base License
  - LAN Base License

Note The LAN Base License is available from Cisco IOS XE Everest 16.5.1.

  - IP Services License
- Connectivity must exist in all network devices.
- Cisco TrustSec SXP software must run on all network devices.

Restrictions for Bidirectional SXP Support

- The peers at each end of the connection must be configured as a bidirectional connection using the **both** keyword. It is a wrong configuration to have one end configured as a bidirectional connection using the **both** keyword and the other end configured as a speaker or listener (unidirectional connection).

Information About Bidirectional SXP Support

Bidirectional SXP Support Overview

Cisco TrustSec builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. The peer that produces data is the speaker and the corresponding peer is the listener.

With the support for bidirectional Security Group Tag (SGT) Exchange Protocol (SXP) configuration, a peer can act as both a speaker and a listener and propagate SXP bindings in both directions using a single connection.

The bidirectional SXP configuration is managed with one pair of IP addresses. On either end, only the listener initiates the SXP connection and the speaker accepts the incoming connection.

Figure 19: Bidirectional SXP Connection

In addition, SXP version 4 (SXPv4) continues to support the loop detection mechanism (to prevent stale binding in the network).

How to Enable Bidirectional SXP Support

Configuring Bidirectional SXP Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp enable**
4. **cts sxp default password**
5. **cts sxp default source-ip**
6. **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} **both** [**vrf** *vrf-name*]
7. **cts sxp speaker hold-time** *minimum-period*

8. **cts sxp listener hold-time** *minimum-period maximum-period*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp enable** **Example:** `Device(config)# cts sxp enable`	Enables the Cisco TrustSec Security Group Tag (SGT) Exchange Protocol version 4 (SXPv4) on a network device.
Step 4	**cts sxp default password** **Example:** `Device(config)# cts sxp default password Cisco123`	(Optional) Specifies the Cisco TrustSec SGT SXP default password.
Step 5	**cts sxp default source-ip** **Example:** `Device(config)# cts sxp default source-ip 10.20.2.2`	(Optional) Configures the Cisco TrustSec SGT SXP source IPv4 address.
Step 6	**cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} **both** [**vrf** *vrf-name*] **Example:** `Device(config)# cts sxp connection peer 10.20.2.2 password default mode local both`	Configures the Cisco TrustSec SXP peer address connection for a bidirectional SXP configuration. The **both** keyword configures the bidirectional SXP configuration. The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the default source address, if configured, or the address of the port. The **password** keyword specifies the password that Cisco TrustSec SXP uses for the connection using the following options: • **default**—Use the default Cisco TrustSec SXP password you configured using the **cts sxp default password** command. • **none**—A password is not used.

	Command or Action	Purpose
		The **mode** keyword specifies the role of the remote peer device: • **local**—The specified mode refers to the local device. • **peer**—The specified mode refers to the peer device. • **both**—Specifies that the device is both the speaker and the listener in the bidirectional SXP connection. The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.
Step 7	**cts sxp speaker hold-time** *minimum-period* **Example:** `Device(config)# cts sxp speaker hold-time 950`	(Optional) Configures the global hold time (in seconds) of a speaker network device for Cisco TrustSec SGT SXPv4. The valid range is from 1 to 65534. The default is 120.
Step 8	**cts sxp listener hold-time** *minimum-period maximum-period* **Example:** `Device(config)# cts sxp listener hold-time 750 1500`	(Optional) Configures the global hold time (in seconds) of a listener network device for Cisco TrustSec SGT SXPv4. The valid range is from 1 to 65534. The default is 90 to 180. **Note** The *maximum-period* value must be greater than or equal to the *minimum-period* value.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Verifying Bidirectional SXP Support Configuration

SUMMARY STEPS

1. **enable**
2. **show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*]

Displays Cisco TrustSec Exchange Protocol (SXP) status and connections.

Example:

```
Device# show cts sxp connections

SXP : Enabled
Highest Version Supported: 4
Default Password : Set
Default Source IP: Not Set
Connection retry open period: 120 secs
Reconcile period: 120 secs
Retry open timer is running
------------------------------------------------
Peer IP : 2.0.0.2
Source IP : 1.0.0.2
Conn status : On (Speaker) :: On (Listener)
Conn version : 4
Local mode : Both
Connection inst# : 1
TCP conn fd : 1(Speaker) 3(Listener)
TCP conn password: default SXP password
Duration since last state change: 1:03:38:03 (dd:hr:mm:sec) :: 0:00:00:46 (dd:hr:mm:sec)


Device# show cts sxp connection brief

SXP : Enabled
Highest Version Supported: 4
Default Password : Set
Default Source IP: Not Set
Connection retry open period: 120 secs
Reconcile period: 120 secs
Retry open timer is running
-----------------------------------------------------
Peer_IP Source_IP Conn Status Duration
-----------------------------------------------------
2.0.0.2 1.0.0.2 On(Speaker)::On(Listener) 0:00:37:17 (dd:hr:mm:sec)::0:00:37:19 (dd:hr:mm:sec)
```

The following table describes the various scenarios for the connection status output.

Table 121: Connection Status Output Scenarios

Node1	Node2	Node1 CLI Output for Connection Status	Node2 CLI Output for Connection Status
Both	Both	On (Speaker) On (Listener)	On (Speaker) On (Listener)
Speaker	Listener	On	On
Listener	Speaker	On	On

Configuration Examples for Bidirectional SXP Support

Example: Configuring Bidirectional SXP Support

The following example shows how to enable bidirectional CTS-SXP and configure the SXP peer connection on Device_A to connect to Device_B:

```
Device_A> enable
Device_A# configure terminal
Device_A(config)# cts sxp enable
Device_A(config)# cts sxp default password Cisco123
Device_A(config)# cts sxp default source-ip 10.10.1.1
Device_A(config)# cts sxp connection peer 10.20.2.2 password default mode local both
Device_A(config)# exit
```

The following example shows how to configure the bidirectional CTS-SXP peer connection on Device_B to connect to Device_A:

```
Device_B> enable
Device_B# configure terminal
Device_B(config)# cts sxp enable
Device_B(config)# cts sxp default password Password123
Device_B(config)# cts sxp default source-ip 10.20.2.2
Device_B(config)# cts sxp connection peer 10.10.1.1 password default mode local both
Device_B(config)# exit
```

Additional References for Bidirectional SXP Support

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Cisco TrustSec configuration	"Cisco TrustSec Support for IOS" chapter in the *Cisco TrustSec Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

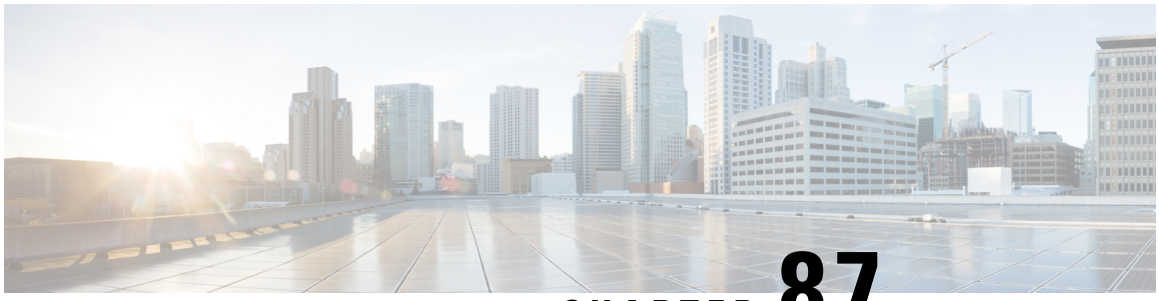
Feature Information for Bidirectional SXP Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 122: Feature Information for Bidirectional SXP Support

Feature Name	Releases	Feature Information
Bidirectional SXP Support		The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection. • Cisco Catalyst 3750-X Series Switches • Cisco Catalyst 3560-X Series Switches • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E • Cisco Catalyst 4500-X Series Switches • Cisco Catalyst 4500E Supervisor Engine 8-E • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3650 Series Switches The following command was introduced or modified: **cts sxp connection peer**.



CHAPTER 87

Cisco TrustSec Interface-to-SGT Mapping

The Cisco TrustSec Interface-to-SGT Mapping feature binds all traffic on a Layer 3 ingress interface to a security group tag (SGT). Once this mapping is implemented, Cisco TrustSec can use the SGT to segregate traffic from various logical Layer 3 ingress interfaces.

Information About Cisco TrustSec Interface-to-SGT Mapping

Interface-to-SGT Mapping

The mapping between interfaces and security group tags (SGTs) is used to map SGTs to traffic of any of the following logical Layer 3 ingress interfaces, regardless of the underlying physical interface:

- Layer 3 (routed) Ethernet interfaces
- Layer 3 (routed) Ethernet 802.1Q subinterfaces
- Tunnel interfaces

The configured SGT tag is assigned to all traffic on the Layer 3 ingress interface and can be used for inline tagging and policy enforcement.

Binding Source Priorities

Cisco TrustSec resolves conflicts among IP address to security group tag (IP-SGT) binding sources with a strict priority scheme. The current priority enforcement order, from lowest to highest, is as follows:

1. CLI—Bindings configured using the **cts role-based sgt-map sgt** command.
2. L3IF—Bindings added due to FIB forwarding entries that have paths through one or more interfaces with consistent Layer 3 Interface to SGT (L3IF-SGT) mapping or identity port mapping on routed ports.
3. SXP—Bindings learned from SGT Exchange Protocol (SXP) peers.

4. INTERNAL—Bindings between locally configured IP addresses and the devices own SGT.

How to Configure Cisco TrustSec Interface-to-SGT Mapping

Configuring Layer 3 Interface-to-SGT Mapping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **cts role-based sgt-map sgt** *sgt-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface gigabitEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 4	**cts role-based sgt-map sgt** *sgt-number* **Example:** `Device(config-if)# cts role-based sgt-map sgt 77`	An SGT is imposed on ingress traffic to the specified interface. • *sgt-number*—Specifies the security group tag (SGT) number. Valid values are from 2 to 65519.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Layer 3 Interface-to-SGT Mapping

SUMMARY STEPS

1. **enable**

2. **show cts role-based sgt-map all**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show cts role-based sgt-map all**

Displays the security group tag (SGT) mapping for the ingress traffic on the Layer 3 interface.

Example:

The following sample output from the **show cts role-based sgt-map all** command shows that once the Cisco TrustSec Interface-to-SGT Mapping feature is implemented, the traffic on the ingress interface is tagged appropriately with Layer 3 interface (L3IF). The output displays the priority scheme of the IP address to security group tag (IP-SGT) binding sources (for more information about the IP-SGT binding source priorities, see the "Binding Source Priorities" section).

```
Device# show cts role-based sgt-map all

IP Address              SGT     Source
============================================
192.0.2.1               4       INTERNAL
192.0.2.5/24            3       L3IF
192.0.2.10/8            3       L3IF
192.0.2.20              5       CLI
198.51.100.1            4       INTERNAL
IP-SGT Active Bindings Summary
============================================
Total number of CLI      bindings = 1
Total number of L3IF     bindings = 2
Total number of INTERNAL bindings = 2
Total number of active   bindings = 5
```

Configuration Examples for Cisco TrustSec Interface-to-SGT Mapping

Example: Configuring Layer 3 Interface-to-SGT Mapping

The following example shows the security group tag (SGT) mapping configuration for the Layer 3 ingress interface:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitEthernet 0/0
Device(config-if)# cts role-based sgt-map sgt 77
Device(config-if)# end
```

Additional References for Cisco TrustSec Interface-to-SGT Mapping

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Cisco TrustSec and SXP configuration	Cisco TrustSec Switch Configuration Guide

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

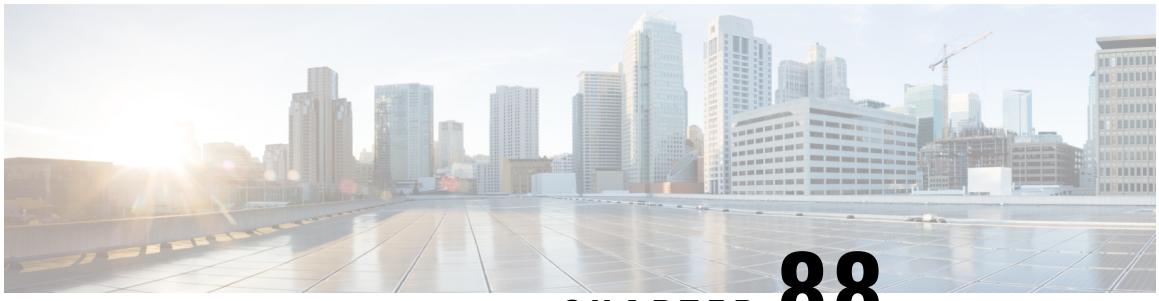
Feature Information for Cisco TrustSec Interface-to-SGT Mapping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 123: Feature Information for Cisco TrustSec Interface-to-SGT Mapping

Feature Name	Releases	Feature Information
Cisco TrustSec Interface-to-SGT Mapping		The Cisco TrustSec Interface-to-SGT Mapping feature binds all traffic on a Layer 3 ingress interface to a security group tag (SGT). Once this mapping is implemented, Cisco TrustSec can use the SGT to segregate traffic from various logical Layer 3 ingress interfaces. The following command was introduced or modified: **cts role-based sgt-map sgt**.



CHAPTER 88

Cisco TrustSec Subnet to SGT Mapping

Subnet to security group tag (SGT) mapping binds an SGT to all host addresses of a specified subnet. Once this mapping is implemented, Cisco TrustSec imposes the SGT on any incoming packet that has a source IP address which belongs to the specified subnet.

Restrictions for Cisco TrustSec Subnet to SGT Mapping

- An IPv4 subnetwork with a /31 prefix cannot be expanded.
- Subnet host addresses cannot be bound to SGTs when the **cts sxp mapping network-map** command *bindings* argument is less than the total number of subnet hosts in the specified subnets or when the number of bindings is 0.
- IPv6 expansions and propagation only occurs when SXP speaker and listener are running SXPv3, or more recent versions.

Information About Cisco TrustSec Subnet to SGT Mapping

In IPv4 networks, SXPv3, and more recent versions, can receive and parse subnet network address/prefix strings from SXPv3 peers. Earlier SXP versions convert the subnet prefix into its set of host bindings before exporting them to an SXP listener peer.

For example, the IPv4 subnet 198.1.1.0/29 is expanded as follows (only 3 bits for host addresses):

- Host addresses 198.1.1.1 to 198.1.1.7 are tagged and propagated to SXP peer.
- Network and broadcast addresses 198.1.1.0 and 198.1.1.8 are not tagged and not propagated.

Note To limit the number of subnet bindings SXPv3 can export, use the **cts sxp mapping network-map** global configuration command.

Subnet bindings are static, which means that active hosts are not learned. They can be used locally for SGT imposition and SGACL enforcement. Packets tagged by subnet to SGT mapping can be propagated on Layer 2 or Layer 3 TrustSec links.

Note For IPv6 networks, SXPv3 cannot export subnet bindings to SXPv2 or SXPv1 peers.

How to Configure Cisco TrustSec Subnet to SGT Mapping

Configuring Subnet to SGT Mapping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp mapping network-map** *bindings*
4. **cts role-based sgt-map** *ipv4-address* **sgt** *number*
5. **cts role-based sgt-map** *ipv6-address::prefix* **sgt** *number*
6. **exit**
7. **show running-config** | **include** *search-string*
8. **show cts sxp connections**
9. **show cts sxp sgt-map**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp mapping network-map** *bindings* **Example:** `Device(config)# cts sxp mapping network-map 10000`	Configures the subnet to SGT mapping host count constraint. The *bindings* argument specifies the maximum number of subnet IP hosts from 0 to 65,535 that can be

	Command or Action	Purpose
		bound to SGTs and exported to the SXP listener. The default is 0 (no expansions performed).
Step 4	**cts role-based sgt-map** *ipv4-address* **sgt** *number* **Example:** `Device(config)# cts role-based sgt-map 10.10.10.10/29 sgt 1234`	(IPv4) Specifies an IPv4 subnet in CIDR notation. The number of bindings specified in step 3 should match or exceed the number of host addresses in the subnet (excluding network and broadcast addresses). The **sgt** *number* keyword pair specifies the SGT number that is to be bound to every host address in the specified subnet. • *ipv4-address*—Specifies the IPv4 network address in dotted decimal notation. • *prefix*—(0 to 30). Specifies the number of bits in the network address. • **sgt** *number* (0-65,535). Specifies the SGT number.
Step 5	**cts role-based sgt-map** *ipv6-address::prefix* **sgt** *number* **Example:** `Device(config)# cts role-based sgt-map 2020::/64 sgt 1234`	(IPv6) Specifies an IPv6 subnet in hexadecimal notation. The number of bindings specified in step 3 should match or exceed the number of host addresses in the subnet (excluding network and broadcast addresses). The **sgt** *number* keyword pair specifies the SGT number that is to be bound to every host address in the specified subnet. • *ipv6-address*—Specifies the IPv4 network address in dotted decimal notation. • *prefix*—(0 to 30). Specifies the number of bits in the network address. • **sgt** *number*—(0-65,535). Specifies the SGT number.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Step 7	**show running-config** | **include** *search-string*
	Example:
	`Device# show running-config | include sgt 1234`
	`Device# show running-config | include network-map`
		Verifies that the **cts role-based sgt-map** and the **cts sxp mapping network-map** commands are in the running configuration.

Step 8	**show cts sxp connections** **Example:** `Device# show cts sxp connections`	Displays the SXP speaker and listener connections with their operational status.
Step 9	**show cts sxp sgt-map** **Example:** `Device# show cts sxp sgt-map`	Displays the IP to SGT bindings exported to the SXP listeners.

	Command or Action	Purpose
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Cisco TrustSec Subnet to SGT Mapping: Examples

The following example shows how to configure IPv4 Subnet to SGT Mapping between two devices running SXPv3 (Device 1 and Device 2):

Configure SXP speaker/listener peering between Device 1 (10.1.1.1) and Device 2 (10.2.2.2).

```
Device1# configure terminal
Device1(config)# cts sxp enable
Device1(config)# cts sxp default source-ip 10.1.1.1
Device1(config)# cts sxp default password 1syzygy1
Device1(config)# cts sxp connection peer 10.2.2.2 password default mode local speaker
```

Configure Device 2 as SXP listener of Device 1.

```
Device2(config)# cts sxp enable
Device2(config)# cts sxp default source-ip 10.2.2.2
Device2(config)# cts sxp default password 1syzygy1
Device2(config)# cts sxp connection peer 10.1.1.1 password default mode local listener
```

On Device 2, verify that the SXP connection is operating:

```
Device2# show cts sxp connections brief | include 10.1.1.1

     10.1.1.1          10.2.2.2          On                3:22:23:18 (dd:hr:mm:sec)
```

Configure the subnetworks to be expanded on Device 1.

```
Device1(config)# cts sxp mapping network-map 10000
Device1(config)# cts role-based sgt-map 10.10.10.0/30 sgt 101
Device1(config)# cts role-based sgt-map 10.11.11.0/29 sgt 11111
Device1(config)# cts role-based sgt-map 172.168.1.0/28 sgt 65000
```

On Device 2, verify the subnet to SGT expansion from Device 1. There should be two expansions for the 10.10.10.0/30 subnetwork, six expansions for the 10.11.11.0/29 subnetwork, and 14 expansions for the 172.168.1.0/28 subnetwork.

```
Device2# show cts sxp sgt-map brief | include 101|11111|65000

IPv4,SGT: <10.10.10.1 , 101>
IPv4,SGT: <10.10.10.2 , 101>
IPv4,SGT: <10.11.11.1 , 11111>
IPv4,SGT: <10.11.11.2 , 11111>
IPv4,SGT: <10.11.11.3 , 11111>
IPv4,SGT: <10.11.11.4 , 11111>
IPv4,SGT: <10.11.11.5 , 11111>
IPv4,SGT: <10.11.11.6 , 11111>
IPv4,SGT: <172.168.1.1 , 65000>
IPv4,SGT: <172.168.1.2 , 65000>
IPv4,SGT: <172.168.1.3 , 65000>
IPv4,SGT: <172.168.1.4 , 65000>
IPv4,SGT: <172.168.1.5 , 65000>
IPv4,SGT: <172.168.1.6 , 65000>
IPv4,SGT: <172.168.1.7 , 65000>
```

```
IPv4,SGT: <172.168.1.8 , 65000>
IPv4,SGT: <172.168.1.9 , 65000>
IPv4,SGT: <172.168.1.10 , 65000>
IPv4,SGT: <172.168.1.11 , 65000>
IPv4,SGT: <172.168.1.12 , 65000>
IPv4,SGT: <172.168.1.13 , 65000>
IPv4,SGT: <172.168.1.14 , 65000>
```

Verify the expansion count on Device 1:

```
Device1# show cts sxp sgt-map

IP-SGT Mappings expanded:22
There are no IP-SGT Mappings
```

Save the configurations on Device 1 and Device 2 and exit global configuration mode.

```
Device1(config)# copy running-config startup-config
Device1(config)# exit

Device2(config)# copy running-config startup-config
Device2(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Cisco TrustSec and SXP configuration	Cisco TrustSec Switch Configuration Guide
IPsec configuration	Configuring Security for VPNs with IPsec
IKEv2 configuration	Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site
Cisco Secure Access Control Server	Configuration Guide for the Cisco Secure ACS

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

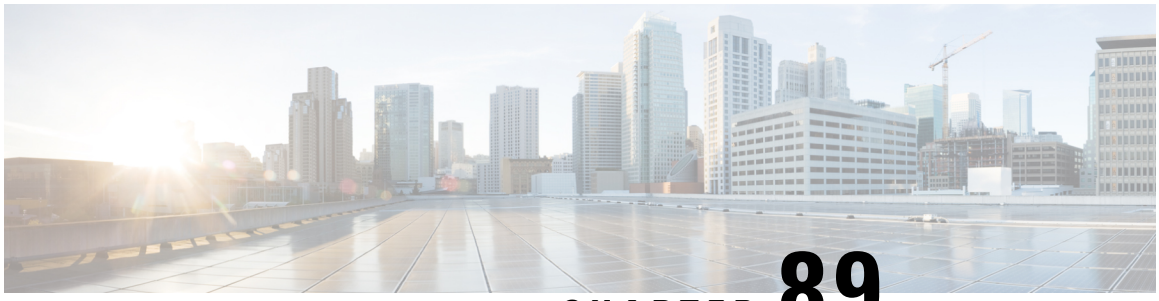
Feature Information for Cisco TrustSec Subnet to SGT Mapping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 124: Feature Information for Cisco TrustSec Subnet to SGT Mapping

Feature Name	Releases	Feature Information
Cisco TrustSec Subnet to SGT Mapping		Subnet to security group tag (SGT) mapping binds an SGT to all host addresses of a specified subnet. Once this mapping is implemented, Cisco TrustSec imposes the SGT on any incoming packet that has a source IP address which belongs to the specified subnet. The following command was introduced: **cts sxp mapping network-map**.



CHAPTER 89

Flexible NetFlow Export of Cisco TrustSec Fields

The Flexible NetFlow Export of Cisco TrustSec Fields feature supports the Cisco TrustSec fields in the Flexible NetFlow (FNF) flow record and helps to monitor, troubleshoot, and identify non-standard behavior for Cisco TrustSec deployments.

This module describes the interaction between Cisco TrustSec and FNF and how to configure and export Cisco TrustSec fields in the NetFlow Version 9 flow records.

Restrictions for Flexible NetFlow Export of Cisco TrustSec Fields

- The security group tag (SGT) value exported in Flexible NetFlow (FNF) records is zero in the following scenarios:
  - The packet is received with an SGT value of zero from a trusted interface.
  - The packet is received without an SGT.
  - The SGT is not found during the IP-SGT lookup.

Information About Flexible NetFlow Export of Cisco TrustSec Fields

Cisco TrustSec Fields in Flexible NetFlow

The Cisco TrustSec fields, source security group tag (SGT) and destination security group tag (DGT) in the Flexible NetFlow (FNF) flow records help administrators correlate the flow with identity information. It enables network engineers to gain a detailed understanding of the customer use of the network and application

resources. This information can then be used to efficiently plan and allocate access and application resources and to detect and resolve potential security and policy violations.

The Cisco TrustSec fields are supported for ingress and egress FNF and for unicast and multicast traffic.

The following table presents Netflow v9 enterprise specific field types for Cisco TrustSec that are used in the FNF templates for the Cisco TrustSec source and destination source group tags.

ID	Description
CTS_SRC_GROUP_TAG	Cisco Trusted Security Source Group Tag
CTS_DST_GROUP_TAG	Cisco Trusted Security Destination Group Tag

The Cisco TrustSec fields are configured in addition to the existing match fields under the FNF flow record. The following configurations are used to add the Cisco TrustSec flow objects to the FNF flow record as key or non-key fields and to configure the source and destination security group tags for the packet.

- The **match flow cts {source | destination} group-tag** command is configured under the flow record to specify the Cisco TrustSec fields as key fields. The key fields differentiate flows, with each flow having a unique set of values for the key fields. A flow record requires at least one key field before it can be used in a flow monitor.
- The **collect flow cts {source | destination} group-tag** command is configured under flow record to specify the Cisco TrustSec fields as non-key fields. The values in non-key fields are added to flows to provide additional information about the traffic in the flows.

The flow record is then configured under flow monitor and the flow monitor is applied to the interface. To export the FNF data, a flow exporter needs to be configured and then added under the flow monitor.

How to Configure Flexible NetFlow Export of Cisco TrustSec Fields

Configuring Cisco TrustSec Fields as Key Fields in the Flow Record

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **match** {**ipv4** | **ipv6**} **protocol**
5. **match** {**ipv4** | **ipv6**} **source address**
6. **match** {**ipv4** | **ipv6**} **destination address**
7. **match transport source-port**
8. **match transport destination-port**
9. **match flow direction**
10. **match flow cts source group-tag**
11. **match flow cts destination group-tag**

12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record cts-record-ipv4`	Creates a new Flexible NetFlow (FNF) flow record, or modifies an existing FNF flow record, and enters Flexible NetFlow flow record configuration mode.
Step 4	**match** {**ipv4** | **ipv6**} **protocol** **Example:** `Device(config-flow-record)# match ipv4 protocol`	(Optional) Configures the IPv4 protocol or IPv6 protocol as a key field for a flow record.
Step 5	**match** {**ipv4** | **ipv6**} **source address** **Example:** `Device(config-flow-record)# match ipv4 source address`	(Optional) Configures the IPv4 or IPv6 source address as a key field for a flow record.
Step 6	**match** {**ipv4** | **ipv6**} **destination address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	(Optional) Configures the IPv4 or IPv6 destination address as a key field for a flow record.
Step 7	**match transport source-port** **Example:** `Device(config-flow-record)# match transport source-port`	(Optional) Configures the transport source port as a key field for a flow record.
Step 8	**match transport destination-port** **Example:** `Device(config-flow-record)# match transport destination-port`	(Optional) Configures the transport destination port as a key field for a flow record.

	Command or Action	Purpose
Step 9	**match flow direction** **Example:** `Device(config-flow-record)# match flow direction`	(Optional) Configures the direction in which the flow is monitored as a key field.
Step 10	**match flow cts source group-tag** **Example:** `Device(config-flow-record)# match flow cts source group-tag`	Configures the Cisco TrustSec source security group tag (SGT) in the FNF flow record as key fields.
Step 11	**match flow cts destination group-tag** **Example:** `Device(config-flow-record)# match flow cts destination group-tag`	Configures the Cisco TrustSec destination security group tag (DGT) in the FNF flow record as key fields.
Step 12	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.

Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **match** {**ipv4** | **ipv6**} **protocol**
5. **match** {**ipv4** | **ipv6**} **source address**
6. **match** {**ipv4** | **ipv6**} **destination address**
7. **match transport source-port**
8. **match transport destination-port**
9. **collect flow direction**
10. **collect flow cts source group-tag**
11. **collect flow cts destination group-tag**
12. **collect counter packets**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record cts-record-ipv4`	Creates a new Flexible NetFlow (FNF) flow record, or modifies an existing FNF flow record, and enters Flexible NetFlow flow record configuration mode.
Step 4	**match** {**ipv4** | **ipv6**} **protocol** **Example:** `Device(config-flow-record)# match ipv4 protocol`	(Optional) Configures the IPv4 protocol or IPv6 protocol as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 5	**match** {**ipv4** | **ipv6**} **source address** **Example:** `Device(config-flow-record)# match ipv4 source address`	(Optional) Configures the IPv4 or IPv6 source address as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 6	**match** {**ipv4** | **ipv6**} **destination address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	(Optional) Configures the IPv4 or IPv6 destination address as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 7	**match transport source-port** **Example:** `Device(config-flow-record)# match transport source-port`	(Optional) Configures the transport source port as a key field for a flow record.
Step 8	**match transport destination-port** **Example:** `Device(config-flow-record)# match transport destination-port`	(Optional) Configures the transport destination port as a key field for a flow record.
Step 9	**collect flow direction** **Example:** `Device(config-flow-record)# collect flow direction`	(Optional) Configures the flow direction as a non-key field and enables the collection of the direction in which the flow was monitored.

	Command or Action	Purpose
Step 10	**collect flow cts source group-tag** **Example:** `Device(config-flow-record)# collect flow cts source group-tag`	Configures the Cisco TrustSec source security group tag (SGT) in the FNF flow record as non-key fields.
Step 11	**collect flow cts destination group-tag** **Example:** `Device(config-flow-record)# collect flow cts destination group-tag`	Configures the Cisco TrustSec destination security group tag (DGT) in the FNF flow record as non-key fields.
Step 12	**collect counter packets** **Example:** `Device(config-flow-record)# collect counter packets`	(Optional) Configures the number of packets seen in a flow as a non-key field and enables collecting the total number of packets from the flow.
Step 13	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.

Configuring a Flow Exporter

Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

Before you begin

Ensure that you create a flow record. For more information see the “Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record” section and the “Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates a flow exporter or modifies an existing flow exporter, and enters Flexible NetFlow flow exporter configuration mode.
Step 4	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter.
Step 5	**end** **Example:** `Device(config-flow-exporter)# end`	Exits Flexible NetFlow flow exporter configuration mode and returns to privileged EXEC mode.

Configuring a Flow Monitor

Before you begin

To add a flow exporter to the flow monitor for data export, ensure that you create the flow exporter. For more information see the “Configuring a Flow Exporter” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor or modifies an existing flow monitor, and enters Flexible NetFlow flow monitor configuration mode.
Step 4	**record** *record-name* **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 5	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the exporter for the flow monitor.
Step 6	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

Applying a Flow Monitor on an Interface

To activate a flow monitor, the flow monitor must be applied to at least one interface.

Before you begin

Ensure that you create a flow monitor. For more information see the “Configuring a Flow Monitor” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device (config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Flexible NetFlow Export of Cisco TrustSec Fields

SUMMARY STEPS

1. **enable**
2. **show flow record** *record-name*
3. **show flow exporter** *exporter-name*
4. **show flow monitor** *monitor-name*
5. **show flow monitor** *monitor-name* **cache**
6. **show flow interface** *type number*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show flow record** *record-name*

Displays the details of the specified Flexible NetFlow (FNF) flow record.

Example:

```
Device> show flow record cts-recordipv4

flow record cts-recordipv4:
  Description:        User defined
  No. of users:       1
  Total field space:  30 bytes
  Fields:
    match ipv4 protocol
    match ipv4 source address
    match ipv4 destination address
    match transport source-port
    match transport destination-port
    match interface input
    match interface output
    match flow direction
    match flow cts source group-tag
    match flow cts destination group-tag
    collect counter packets
```

Step 3 **show flow exporter** *exporter-name*

Displays the current status of the specified FNF flow exporter.

Example:

```
Device> show flow exporter EXPORTER-1

Flow Exporter EXPORTER-1:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 100.100.100.1
    Source IP address:      3.3.3.2
    Transport Protocol:     UDP
    Destination Port:       2055
    Source Port:            65252
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Used
```

Step 4 **show flow monitor** *monitor-name*

Displays the status and statistics of the specified FNF flow monitor.

Example:

```
Device> show flow monitor FLOW-MONITOR-1

Flow Monitor FLOW-MONITOR-1:
  Description:       User defined
  Flow Record:       cts-recordipv4
  Flow Exporter:     EXPORTER-1
  Cache:
    Type:                normal (Platform cache)
    Status:              allocated
    Size:                200000 entries
    Inactive Timeout:    60 secs
    Active Timeout:      1800 secs
    Update Timeout:      1800 secs
    Synchronized Timeout: 600 secs
    Trans end aging:     off
```

Step 5 **show flow monitor** *monitor-name* **cache**

Displays the contents of the specified FNF flow monitor cache.

Example:

```
Device> show flow monitor FLOW-MONITOR-1 cache

  Cache type:                               Normal
  Cache size:                               4096
  Current entries:                          2
  High Watermark:                           2

  Flows added:                              6
  Flows aged:                               4
    - Active timeout      (1800 secs)       0
    - Inactive timeout    (15 secs)         4
    - Event aged                            0
    - Watermark aged                        0
    - Emergency aged                        0

  IPV4 SOURCE ADDRESS:                      10.1.0.1
  IPV4 DESTINATION ADDRESS:                 172.16.2.0
  TRNS SOURCE PORT:                         58817
  TRNS DESTINATION PORT:                    23
  FLOW DIRECTION:                           Input
  IP PROTOCOL:                              6
  SOURCE GROUP TAG:                         100
  DESTINATION GROUP TAG:                    200
  counter packets:                          10

  IPV4 SOURCE ADDRESS:                      172.16.2.0
  IPV4 DESTINATION ADDRESS:                 10.1.0.1
  TRNS SOURCE PORT:                         23
  TRNS DESTINATION PORT:                    58817
  FLOW DIRECTION:                           Output
  IP PROTOCOL:                              6
  SOURCE GROUP TAG:                         200
  DESTINATION GROUP TAG:                    100
```

```
  counter packets:                          8
```

Step 6 **show flow interface** *type number*

Displays the details of the FNF flow monitor applied on the specified interface. If a flow monitor is not applied on the interface, then the output is empty.

Example:

```
Device>  show flow interface GigabitEthernet0/0/3

Interface GigabitEthernet0/0/3
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Input
        traffic(ip):      on
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Output
        traffic(ip):      on
```

Configuration Examples for Flexible NetFlow Export of Cisco TrustSec Fields

Example: Configuring Cisco TrustSec Fields as Key Fields in the Flow Record

The following example shows how to configure the Cisco TrustSec flow objects as key fields in an IPv4 Flexible NetFlow flow record:

```
Device> enable
Device# configure terminal
Device(config)# flow record cts-record-ipv4
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow direction
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# end
```

Example: Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record

The following example shows how to configure the Cisco TrustSec flow objects as non-key fields in an IPv4 Flexible NetFlow flow record:

```
Device> enable
Device# configure terminal
Device(config)# flow record cts-record-ipv4
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# collect flow direction
Device(config-flow-record)# collect flow cts source group-tag
Device(config-flow-record)# collect flow cts destination group-tag
Device(config-flow-record)# collect counter packets
Device(config-flow-record)# end
```

Example: Configuring a Flow Exporter

```
Device> enable
Device# configure terminal
Device(config)# flow exporter EXPORTER-1
Device(config-flow-exporter)# destination 172.16.10.2
Device(config-flow-exporter)# end
```

Example: Configuring a Flow Monitor

```
Device> enable
Device# configure terminal
Device(config)# flow monitor FLOW-MONITOR-1
Device(config-flow-monitor)# record FLOW-RECORD-1
Device(config-flow-monitor)# exporter EXPORTER-1
Device(config-flow-monitor)# end
```

Example: Applying a Flow Monitor on an Interface

The following example shows how to activate an IPv4 flow monitor by applying it to an interface to analyze traffic. To activate an IPv6 flow monitor, replace the **ip** keyword with the **ipv6** keyword.

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet 0/0
```

```
Device(config-if)# ip flow monitor FLOW-MONITOR-1 input
Device(config-if)# end
```

Additional References for Flexible NetFlow Export of Cisco TrustSec Fields

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Data export in Flexible NetFlow	"Flexible NetFlow Output Features on Data Export" chapter in the *Flexible Netflow Configuration Guide* publication
Flexible NetFlow flow records and flow monitors	"Customizing Flexible NetFlow Flow Records and Flow Monitors" chapter in the *Flexible Netflow Configuration Guide* publication

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

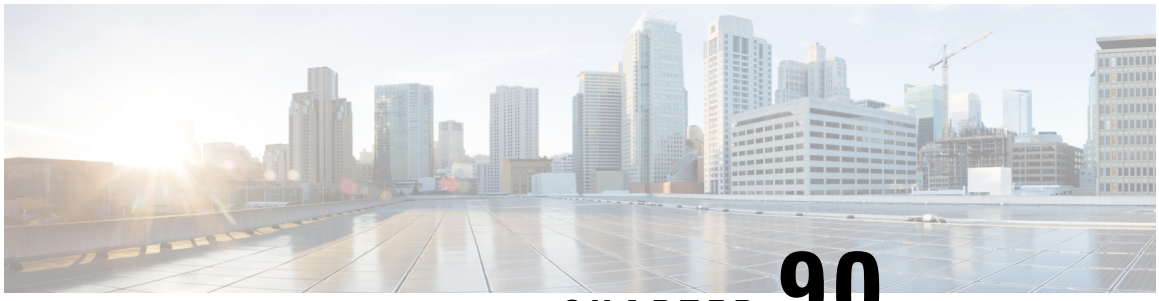
Feature Information for Flexible NetFlow Export of Cisco TrustSec Fields

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 125: Feature Information for Flexible NetFlow Export of Cisco TrustSec Fields

Feature Name	Releases	Feature Information
Flexible NetFlow Export of Cisco TrustSec Fields		The Flexible NetFlow Export of Cisco TrustSec Fields feature supports the Cisco TrustSec fields in the Flexible NetFlow (FNF) flow record and helps to monitor, troubleshoot, and identify non-standard behavior for Cisco TrustSec deployments. The following commands were introduced by this feature: **match flow cts {source | destination} group-tag** and **collect flow cts {source | destination} group-tag**.



CHAPTER 90

Cisco TrustSec SGT Caching

The Cisco TrustSec SGT Caching feature enhances the ability of Cisco TrustSec to make Security Group Tag (SGT) transportability flexible. This feature identifies the IP-SGT binding and caches the corresponding SGT so that network packets are forwarded through all network services for normal deep packet inspection processing and at the service egress point the packets are re-tagged with the appropriate SGT.

Restrictions for Cisco TrustSec SGT Caching

The global Security Group Tag (SGT) caching configuration and the interface-specific ingress configuration are mutually exclusive. In the following scenarios, a warning message is displayed if you attempt to configure SGT caching both globally and on an interface:

- If an interface has ingress SGT caching enabled using the **cts role-based sgt-cache ingress** command in interface configuration mode, and a global configuration is attempted using the **cts role-based sgt-caching** command, a warning message is displayed as shown in this example:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitEthernet0/0
Device(config-if)# cts role-based sgt-cache ingress
Device(config-if)# exit
Device(config)# cts role-based sgt-caching

There is at least one interface that has ingress sgt caching configured. Please remove
 all interface ingress sgt caching configuration(s) before attempting global enable.
```

- If global configuration is enabled using the **cts role-based sgt-caching** command, and an interface configuration is attempted using the **cts role-based sgt-cache ingress** command in interface configuration mode, a warning message is displayed as shown in this example:

```
Device> enable
Device# configure terminal
```

```
Device(config)# cts role-based sgt-caching
Device(config)# interface gigabitEthernet0/0
Device(config-if)# cts role-based sgt-cache ingress

Note that ingress sgt caching is already active on this interface due to global
sgt-caching enable.
```

- SGT Caching for Tunneling of IPv6 packet over V4 transport & IPv4 packet over V6 transport is not supported.
- High availability and syncing of IPv6 SGACL policies on the routing platforms are not supported for IPv6-SGT caching.
- SGT caching is not supported for IPSec packets carrying SGT tags in ESP header on ISR4K based platforms.
- SGT caching is not performed for the link-local IPv6 source address.

  A link-local address is a network address that is valid only for communications within the network segment (link) or the broadcast domain that the host is connected to. Link-local addresses are not guaranteed to be unique beyond a single network segment. Therefore, routers do not forward packets with link-local addresses. Because they are not unique, SGT tags for the packets with source as link-local IPv6 address are not assigned.
- SGT caching is not supported on tunnel interfaces that have IPsec with IVRF configured.
- Configuring SGT caching on a virtual template interface is not supported on a Cisco ASR 1000 platform.

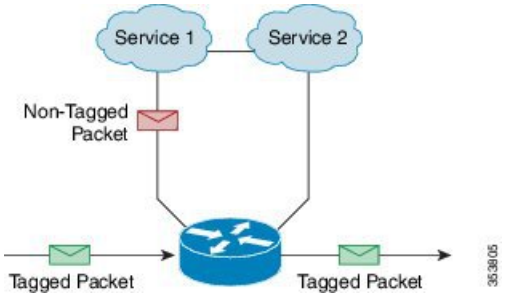
Information About Cisco TrustSec SGT Caching

Identifying and Reapplying SGT Using SGT Caching

Cisco TrustSec uses Security Group Tag (SGT) caching to ensure that traffic tagged with SGT can also pass through services that are not aware of SGTs. Examples of services that cannot propagate SGTs are WAN acceleration or optimization, intrusion prevention systems (IPS), and upstream firewalls. In one-arm mode, a packet tagged with SGT enters a device (where the tags are cached), and is redirected to a service. After that service is completed, the packet either returns to the device, or is redirected to another device as shown in the figure. In such a scenario:

1. The Cisco TrustSec SGT Caching feature enables the device to identify the IP-SGT binding information from the incoming packet and caches this information.
2. The device redirects the packet to the service or services that cannot propagate SGTs.
3. After the completion of the service, the packet returns to the device.
4. The appropriate SGT is reapplied to the packet at the service egress point.
5. Role-based enforcements are applied to the packet that has returned to the device from the service or services.
6. The packet with SGTs is forwarded to other Cisco TrustSec-capable devices downstream.

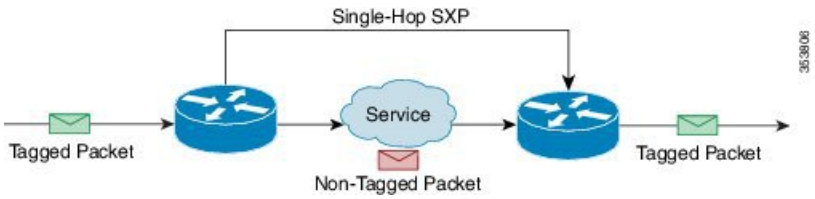
Figure 20: SGT Caching in One-Arm Mode



In certain instances, some services are deployed in a bump-in-the-wire topology. In such a scenario:

1. The packets that go through a service or services do not come back to the device.
2. Single-hop SGT Exchange Protocol (SXP) is used to identify and export the identified IP-SGT bindings.
3. The upstream device in the network identifies the IP-SGT bindings through SXP and reapplies the appropriate tags or uses them for SGT-based enforcement. During egress caching, the original pre-Network Address Translation (NAT) source IP address is cached as part of the identified IP-SGT binding information.
4. IP-SGT bindings that do not receive traffic for 300 seconds are removed from the cache.

Figure 21: SGT Caching in Bump-in-the-wire Topology



SGT Caching for IPv6 Traffic

The following are the considerations for SGT caching for IPv6 traffic:

- **Global Unicast IPv6 Packet**: IPv6-SGT caching is performed for traffic coming in ingress and egress directions for IPv6 packets. The SGT tags come inline in the packet (ethernet header, IPSec header, GRE header). However, SGT caching for tag in IPSec packet is not supported on ISR4K based platforms.
- **Multicast IPv6 Address**: SGT caching is not supported for IPv6 multicast traffic and link local IPv6 addresses.
- **Export of Cached IPv6-SGT Binding Via SXP**: The IPv6-SGT binding learnt in the data-plane is notified to the RBM (RoleBased Manager) database in IOS. These bindings can then be exported to other trustsec devices using the SXP.

How to Configure Cisco TrustSec SGT Caching

Configuring SGT Caching Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts role-based sgt-caching**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts role-based sgt-caching** **Example:** `Device(config)# cts role-based sgt-caching`	Enables SGT caching in ingress direction for all interfaces.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring SGT Caching on an Interface

When an interface is configured to be on a Virtual Routing and Forwarding (VRF) network, the IP-SGT bindings identified on that interface are added under the specific VRF. (To view the bindings identified on a corresponding VRF, use the **show cts role-based sgt-map vrf** *vrf-name* **all** command.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*

4. **cts role-based sgt-cache** [**ingress** | **egress**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface gigabitEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**cts role-based sgt-cache** [**ingress**	**egress**] **Example:** `Device(config-if)# cts role-based sgt-cache ingress`	Configures SGT caching on a specific interface. • ingress—Enables SGT caching for traffic entering the specific interface (inbound traffic). • egress—Enables SGT caching for traffic exiting the specific interface (outbound traffic).
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Cisco TrustSec SGT Caching

SUMMARY STEPS

1. **enable**
2. **show cts**
3. **show cts interface**
4. **show cts interface brief**
5. **show cts role-based sgt-map all ipv4**
6. **show cts role-based sgt-map vrf**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 show cts

Displays Cisco TrustSec connections and the status of global SGT caching.

Example:

```
Device# show cts

Global Dot1x feature: Disabled
CTS device identity: ""
CTS caching support: disabled
CTS sgt-caching global: Enabled
Number of CTS interfaces in DOT1X mode:  0,    MANUAL mode: 0
Number of CTS interfaces in LAYER3 TrustSec mode: 0
Number of CTS interfaces in corresponding IFC state
  INIT            state:  0
  AUTHENTICATING  state:  0
  AUTHORIZING     state:  0
  SAP_NEGOTIATING state:  0
  OPEN            state:  0
  HELD            state:  0
  DISCONNECTING   state:  0
  INVALID         state:  0
CTS events statistics:
  authentication success: 0
  authentication reject : 0
  authentication failure: 0
  authentication logoff : 0
  authentication no resp: 0
  authorization success : 0
  authorization failure : 0
  sap success           : 0
  sap failure           : 0
  port auth failure     : 0
```

Step 3 show cts interface

Displays Cisco TrustSec configuration statistics for an interface and SGT caching information with mode details (ingress or egress).

Example:

```
Device# show cts interface GigabitEthernet0/1

Interface GigabitEthernet0/1
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is enabled, mode:     MANUAL
      Propagate SGT:          Enabled
```

```
      Static Ingress SGT Policy:
        Peer SGT:             200
        Peer SGT assignment:  Trusted

    L2-SGT Statistics
        Pkts In                     : 16298041
        Pkts (policy SGT assigned)  : 0
        Pkts Out                    : 5
        Pkts Drop (malformed packet): 0
        Pkts Drop (invalid SGT)     : 0
```

Step 4 **show cts interface brief**

Displays SGT caching information with mode details (ingress or egress) for all interfaces.

Example:

```
Device# show cts interface brief

Interface GigabitEthernet0/0
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is disabled

Interface GigabitEthernet0/1
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is enabled, mode:     MANUAL
      Propagate SGT:          Enabled
      Static Ingress SGT Policy:
        Peer SGT:             200
        Peer SGT assignment:  Trusted

Interface GigabitEthernet0/2
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is enabled, mode:     MANUAL
      Propagate SGT:          Enabled
      Static Ingress SGT Policy:
        Peer SGT:             0
        Peer SGT assignment:  Untrusted

Interface GigabitEthernet0/3
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is disabled

Interface Backplane-GigabitEthernet0/4
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is disabled

Interface RG-AR-IF-INPUT1
    CTS sgt-caching Ingress:  Enabled
    CTS sgt-caching Egress :  Disabled
    CTS is disabled
```

Step 5 **show cts role-based sgt-map all ipv4**

Displays all the SGT-IPv4 bindings.

Example:

```
Device# show cts role-based sgt-map all ipv4

Active IPv4-SGT Bindings Information

IP Address              SGT     Source
============================================
192.0.2.1                50      CACHED
192.0.2.2                50      CACHED
192.0.2.3                50      CACHED
192.0.2.4                50      CACHED
192.0.2.5                3900    INTERNAL
192.0.2.6                3900    INTERNAL
192.0.2.7                3900    INTERNAL

IP-SGT Active Bindings Summary
============================================
Total number of CACHED   bindings = 20
Total number of INTERNAL bindings = 3
Total number of active   bindings = 23
```

Step 6 **show cts role-based sgt-map vrf**

Displays all the SGT-IP bindings for the specific Virtual Routing and Forwarding (VRF) interface.

Example:

```
Device# show cts role-based sgt-map vrf

%IPv6 protocol is not enabled in VRF RED
Active IPv4-SGT Bindings Information

IP Address              SGT     Source
============================================
192.0.2.1                50      CACHED
192.0.2.2                2007    CACHED
192.0.2.3                50      CACHED
192.0.2.4                50      CACHED
```

Verifying IP-to-SGT Bindings

Displays the IP-to-SGT bindings learnt in the data-plane.

```
Device# show cts role-based sgt-map all
Active IPv4-SGT Bindings Information

IP Address              SGT     Source
============================================
10.104.33.219           300     INTERNAL

IP-SGT Active Bindings Summary
============================================
Total number of INTERNAL bindings = 1
Total number of active   bindings = 1

Active IPv6-SGT Bindings Information
```

```
IP Address                                SGT     Source
============================================================
100::/64                                  124     CLI
200::2                                    300     INTERNAL
300::1                                    300     INTERNAL
1000::2                                   300     INTERNAL

IP-SGT Active Bindings Summary
============================================
Total number of CLI      bindings = 1
Total number of INTERNAL bindings = 3
Total number of active   bindings = 4
```

Configuration Examples for Cisco TrustSec SGT Caching

Example: Configuring SGT Caching Globally

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-caching
Device(config)# end
```

Example: Configuring SGT Caching for an Interface

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitEthernet 0/1/0
Device(config-if)# cts role-based sgt-cache ingress
Device(config-if)# end
```

Example: Disabling SGT Caching on an Interface

The following example shows how to disable SGT caching on an interface and displays the status of SGT caching on the interface when caching is enabled globally, but disabled on the interface.

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-caching
Device(config)# interface gigabitEthernet 0/1
Device(config-if)# no cts role-based sgt-cache ingress
Device(config-if)# end
Device# show cts interface GigabitEthernet0/1

Interface GigabitEthernet0/1
    CTS sgt-caching Ingress:  Disabled
    CTS sgt-caching Egress :  Disabled
```

```
CTS is enabled, mode:     MANUAL
  Propagate SGT:          Enabled
  Static Ingress SGT Policy:
    Peer SGT:             200
    Peer SGT assignment:  Trusted

L2-SGT Statistics
    Pkts In                     : 200890684
    Pkts (policy SGT assigned)  : 0
    Pkts Out                    : 14
    Pkts Drop (malformed packet): 0
    Pkts Drop (invalid SGT)     : 0
```

Additional References for Cisco TrustSec SGT Caching

Related Documents

Related Topic	Document Title
Cisco IOS Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Cisco TrustSec configuration	"Cisco TrustSec Support for IOS" chapter in the *Cisco TrustSec Configuration Guide*
Cisco TrustSec overview	Overview of TrustSec
Cisco TrustSec solution	Cisco TrustSec Security Solution

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

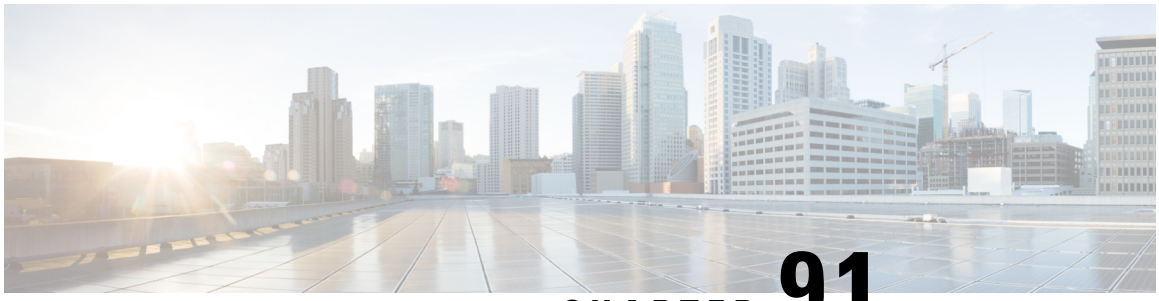
Feature Information for Cisco TrustSec SGT Caching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 126: Feature Information for Cisco TrustSec SGT Caching

Feature Name	Releases	Feature Information
Cisco TrustSec SGT Caching		The Cisco TrustSec SGT Caching feature enhances the ability of Cisco TrustSec to make Security Group Tag (SGT) transportability flexible. This feature identifies the IP-SGT binding and caches the corresponding SGT so that network packets are forwarded through all network services for normal deep packet inspection processing and at the service egress point the packets are re-tagged with the appropriate SGT. The following commands were introduced or modified: **cts role-based sgt-caching**, **cts role-based sgt-cache** [**ingress** | **egress**].
IPv6 enablement - SGT Caching	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.



CHAPTER 91

CTS SGACL Support

CTS SGACL support feature provides state-less access control mechanism based on the security association or security group tag value instead of IP addresses.

Prerequisites for CTS SGACL Support

For CTS SGACL support, ensure that Protected Access Credential (PAC) and environmental data download is configured on the device for dynamic SGACL.

Restrictions for CTS SGACL Support

- For the list of supported TrustSec features per platform and the minimum required IOS release, see the Cisco TrustSec Platform Support Matrix at the following URL: http://www.cisco.com/en/US/solutions/ns170/ns896/ns1051/trustsec_matrix.html
- SGACL enforcement is not supported on management interfaces.
- Dynamic SGACL download size is limited to 6 KB
- There is no validation of SGACL enforcement on Port-Channel interfaces.
- In a VRF aware SGT configuration, Cisco IOS XE Denali 16.3 supports ISE communication though non management VRF interface. ISE communication through management interface is not supported.
- Scale limit of 6 KB is only for dynamic SGACL. Static SGACL can support higher scale like 256*256 matrix.
- SGACL enforcement is by-passed for the IPv6 packets with link-local IPv6 source/destination address.
- The SGACL enforcement for IPv6 multicast traffic is by-passed.

- Starting with Cisco IOS XE Bengaluru 17.4.1, you can configure automated tester to be VRF aware. You can use the **vrf** keyword with the **automate-tester** command to enable automate-tester for a non-default VRF.

Note For VRF aware automate-tester to work, you must configure the **global config ipv4/ipv6 source interface** *interface-name* **vrf** *vrf-name* command.

Information About CTS SGACL Support

CTS SGACL Support

Security group access control lists (SGACLs) is a policy enforcement through which the administrator can control the operations performed by the user based on the security group assignments and destination resources. Policy enforcement within the Cisco Trustsec domain is represented by a permissions matrix, with source security group number on one axis and destination security group number on the other axis. Each cell in the matrix contains an ordered list of SGACLs which specifies the permissions that should be applied to packets originating from an IP belonging to a source security group and having a destination IP that belongs to the destination security group.

SGACL provides state-less access control mechanism based on the security association or security group tag value instead of IP addresses and filters the traffic based on match class. There are three ways to provision the SGACL policy:

- Static policy provisioning - The SGACL policies are defined by the user using the command **cts role-based permission**.
- Dynamic policy provisioning - Configuration of SGACL policies should be done primarily through the policy management function of the Cisco Secure ACS or the Cisco Identity Services Engine - Cisco Identity Services Engine User Guide
- Change of Authorization (CoA) - The updated policy is downloaded when the SGACL policy is modified on the ISE and CoA is pushed to the CTS device.

SGACL Monitor Mode

During the pre-deployment phase of Cisco TrustSec, an administrator will use the monitor mode to test the security policies without enforcing them to make sure that the policies function as intended. If the security policies do not function as intended, the monitor mode provides a convenient mechanism for identifying that and provides an opportunity to correct the policy before enabling SGACL enforcement. This enables administrators to have increased visibility to the outcome of the policy actions before they enforce it, and confirm that the subject policy meets the security requirements (access is denied to resources if users are not authorized).

The monitoring capability is provided at the SGT-DGT pair level. When you enable the SGACL monitoring mode feature, the deny action is implemented as an ACL permit on the line cards. This allows the SGACL counters and logging to display how connections are handled by the SGACL policy. Since all the monitored traffic is permitted, there is no disruption of service due to SGACLs while in the SGACL monitor mode.

How to Configure CTS SGACL Support

Enabling SGACL Policy Enforcement Globally

To enable SGACL policy enforcement on Cisco TrustSec-enabled routed interfaces, perform this task:

```
enable
configure terminal
cts role-based enforcement
```

Enabling SGACL Policy Enforcement Per Interface

You can enable SGACL enforcement globally and disable on a specific interface with **cts role-based enforcement** command. SGACL enforcement can also be enabled on specific interfaces without enabling it globally.

To enable SGACL policy enforcement on interfaces, perform this task:

```
enable
configure terminal
interface GigabitEthernet 0/1/1
cts role-based enforcement
```

Configuring IPv6 SGACL Access Control Entries

An SGACL is defined similar to the extended named ACL using the following command:

```
Device(config)#ipv6 access-list role-based sgacl1
IPV6 Role-based Access List Configuration commands:
  default   Set a command to its defaults
  deny      Specify packets to reject
  exit      Exit from access-list configuration mode
  no        Negate a command or set its defaults
  permit    Specify packets to forward
  remark    Access list entry comment
  sequence  Sequence number for this entry
```

Attaching SGACLs to Permission Matrix Cell

```
Device(config)#cts role-based permissions from 100 to 200
  WORD  Role-based Access-list name
  ipv4  Protocol Version - IPv4
  ipv6  Protocol Version - IPv6
```

This command defines, replaces, or deletes the list of RBACLs for a given <SGT, DGT> pair. This policy comes into an effect when there is no dynamic policy for the same SGT, DGT. By default, you can attach only an IPv4 type RBACL. To add an IPv6 SGACL, specify **ipv6** explicitly.

Manually Configuring SGACL Policies

To manually configure SGACL policies, perform the following tasks:

```
enable
configure terminal
ip access-list role-based allow_webtraff
10 permit tcp dst eq 80
20 permit tcp dst eq 443
cts role-based permissions from 55 to 66 allow_webtraff
end
```

Refreshing the Downloaded SGACL Policies

To refresh the downloaded SGACL policies, perform the following task:

```
enable
cts refresh policy
```

Or

```
enable
cts refresh policy sgt 10
```

Configuring SGACL Monitor Mode

Before configuring SGACL monitor mode, ensure that Cisco TrustSec is enabled.

Note The device level monitor mode is not enabled by default unless any one of the configurations are applied. In case of SGACL's downloaded from ISE, the monitor mode state from ISE takes precedence always. This is applicable for both per-cell monitor mode or global monitor mode which is applicable for all cell.

```
configure terminal
cts role-based monitor enable
cts role-based monitor permissions from 2 to 3 ipv4
show cts role-based permissions from 2 to 3 ipv4
show cts role-based counters ipv4
```

Configuring IPv6 SGACL ACE

The following CLI is used to define Access Control Entries (ACEs) of an IPv6 SGACL.

```
Device(config)#ipv6 access-list role-based sgacl1
Device(config-ipv6rb-acl)#permit ipv6
Device(config-ipv6rb-acl)#exit
Device(config)#cts role-based permissions from 100 to 200 ipv6 sgacl1
```

Note IPv6 ACL configuration is for static SGACL whereas for dynamic SGACL, ACEs are configured on the ISE.

Configuration Examples for CTS SGACL Support

Example: CTS SGACL Support

The following is a sample output of the show cts role-based permissions command.

```
Router# show cts role-based permissions

IPv4 Role-based permissions default:
        default_sgacl-02
        Permit IP-00
IPv4 Role-based permissions from group 55:SGT_55 to group 66:SGT_66 (configured):
        allow_webtraff
RBACL Monitor All for Dynamic Policies : FALSE
RBACL Monitor All for Configured Policies : FALSE


Router#sh cts role-based permissions ipv6
IPv6 Role-based permissions from group 2103:Cisco_UC_Servers to group 2104:Exchange_Servers:

        SGACL_5-10-ipv6
RBACL Monitor All for Dynamic Policies : FALSE
RBACL Monitor All for Configured Policies : FALSE
```

The following is a sample output, applicable only to dynamic SGACL, of the show cts policy sgt command.

```
Router# show cts policy sgt

CTS SGT Policy
==============
RBACL Monitor All : FALSE
RBACL IP Version Supported: IPv4
SGT: 0-02:Unknown
SGT Policy Flag: 0xc1408801
RBACL Source List: Empty
RBACL Destination List: Not exist
RBACL Multicast List: Not exist
RBACL Policy Lifetime = 1800 secs
RBACL Policy Last update time = 20:58:28 IST Wed Jul 13 2016
Policy expires in 0:00:24:05 (dd:hr:mm:sec)
Policy refreshes in 0:00:24:05 (dd:hr:mm:sec)
Cache data applied = NONE

SGT: 65535-46:ANY
SGT Policy Flag: 0x41400001
RBACL Source List:
  Source SGT: 65535-46:ANY-0, Destination SGT: 65535-46:ANY-0
  rbacl_type = 80
```

```
  rbacl_index = 1
  name   = default_sgacl-02
  IP protocol version = IPV4
  refcnt = 1
  flag   = 0x40000000
  stale  = FALSE
  RBACL ACEs:
    permit icmp
    permit ip
  Source SGT: 65535-46:ANY-0, Destination SGT: 65535-46:ANY-0
  rbacl_type = 80
  rbacl_index = 2
  name   = Permit IP-00
  IP protocol version = IPV4
  refcnt = 1
  flag   = 0x40000000
  stale  = FALSE
  RBACL ACEs:
    permit ip
RBACL Destination List: Not exist
RBACL Multicast List: Not exist
RBACL Policy Lifetime = 1800 secs
RBACL Policy Last update time = 20:58:28 IST Wed Jul 13 2016
Policy expires in 0:00:24:05 (dd:hr:mm:sec)
Policy refreshes in 0:00:24:05 (dd:hr:mm:sec)
Cache data applied = NONE
```

The following is a sample output, applicable only to dynamic SGACL, of the show cts rbacl command.

```
Router# show cts rbacl

CTS RBACL Policy
================
RBACL IP Version Supported: IPv4 & IPv6
  name   =multple_ace-16
  IP protocol version = IPV4
  refcnt = 4
  flag = 0x40000000
  stale = FALSE
  RBACL ACEs:
      permit icmp
      deny tcp

  name   =default_sgacl-02
  IP protocol version = IPV4
  refcnt = 2
  flag = 0x40000000
  stale = FALSE
  RBACL ACEs:
      permit icmp
      permit ip

  name   =SGACL_256_ACE-71
  IP protocol version = IPV4
```

Example: Configuring SGACL Monitor Mode

The following is a sample configuration example for SGACL Monitor Mode:

```
Device# configure terminal
Device(config)# cts role-based monitor enable
Device(config)# cts role-based permissions from 2 to 3 ipv4
Device# show cts role-based permissions from 2 to 3 ipv4

        IPv4 Role-based permissions from group 2:sgt2 to group 3:sgt3 (monitored):
        denytcpudpicmp-10
        Deny IP-00

Device# show cts role-based permissions from 2 to 3 ipv4 details

        IPv4 Role-based permissions from group 2:sgt2 to group 3:sgt3 (monitored):
        denytcpudpicmp-10
        Deny IP-00
        Details:
        Role-based IP access list denytcpudpicmp-10 (downloaded)
        10 deny tcp
        20 deny udp
        30 deny icmp
        Role-based IP access list Permit IP-00 (downloaded)
        10 permit ip
Device# show cts role-based permissions ipv6
IPv6 Role-based permissions from group 201 to group 22 (configured):
        g6
IPv6 Role-based permissions from group 100 to group 200 (configured):
        sgacl1
RBACL Monitor All for Dynamic Policies : FALSE
RBACL Monitor All for Configured Policies : FALSE

Device# show cts role-based counters ipv4
Role-based IPv4 counters
From    To      SW-Denied  HW-Denied  SW-Permitt HW-Permitt SW-Monitor HW-Monitor
100     200     0          0          0          0          0          0
101     201     0          0          0          0          0          0

Device# show cts role-based counters ipv6
Role-based IPv6 counters
From    To      SW-Denied  HW-Denied  SW-Permitt HW-Permitt SW-Monitor HW-Monitor
201     22      0          0          0          0          0          0
100     200     0          0          0          0          0          0
```

Example: Refreshing the Downloaded SGACL Policies

The following is a sample configuration example for refreshing the downloaded SGACL policies. The command is run in a privileged EXEC mode.

```
Router#cts refresh policy
Router#cts refresh policy sgt
```

Additional References for CTS SGACL Support

Related Documents

MIBs

MIB	MIBs Link
CISCO-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

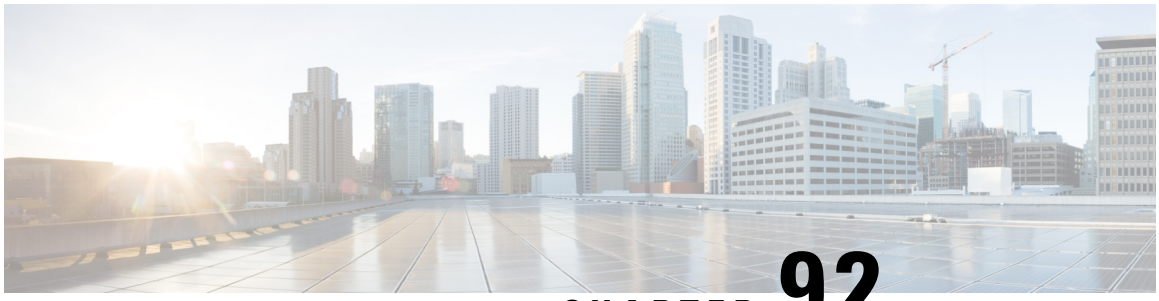
Feature Information for CTS SGACL Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 127: Feature Information for CTS SGACL Support

Feature Name	Releases	Feature Information
CTS SGACL Support	Cisco IOS Release 16.3	The CTS SGACL Support feature provides state-less access control mechanism based on the security association or security group tag value instead of IP addresses. In Cisco IOS Release 16.3, this feature was introduced for Cisco Aggregation Service Router 1000 series and Integrated Services Router 4000 series. The following commands were introduced by this feature: **cts role-based enforcement, ip access-list role-based, cts role-based permissions, show cts role-based permissions, show cts rbacl**.
TrustSec SGACL Monitor Mode	Cisco IOS XE Everest 16.4.1	TrustSec SGACL Monitor Mode feature monitors the security policies without enforcing that the policies function as intended. The monitor mode provides a convenient mechanism for identifying the security policies that do not function and provide an opportunity to correct the policy before enabling SGACL enforcement. The following commands were introduced by this feature: **cts role-based monitor enable, cts role-based monitor permissions**.
IPv6 enablement - SGACL Enforcement	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.



CHAPTER 92

Accessing TrustSec Operational Data Externally

Cisco TrustSec builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. Communication on the links between devices in the domain is secured with a combination of encryption, message integrity check, and data-path replay protection mechanisms.

Cisco TrustSec also provides security using group-based access control - access policies within the Cisco TrustSec domain are topology-independent, and are based on the roles of source and destination devices rather than on network addresses. Individual packets are tagged with the security group number of the source.

Cisco TrustSec produces two kinds of data - namely configuration data and operational data. Configuration data comes from the config programming model and the operational data comes from the operational data model.

It is possible to access TrustSec operational data from external applications that can handle data that is structured using YANG. Using the Netconf and Restconf protocol, the external device is able to extract operational information from Cisco devices - thereby providing programmability over an external interface.

Prerequisites for Accessing Cisco TrustSec Operational Data Externally

- An understanding of Cisco Trustsec, security tag propagation using SXP across network devices, and policy enforcement.
- Effective Cisco IOS XE Everest 16.5.1, Cisco TrustSec supports crypto k9 image with licenses for IP services or IP base only.
- The NETCONF or RESTCONF protocol should be enabled on the Cisco device. To enable the NETCONF protocol, use the command **netconf-yang** in the configuration mode.

Note The LANbase license supports only SXP; SGACL and IP-SGT operational data are not supported.

Restrictions for Accessing Cisco TrustSec Operational Data Externally

- Operation data limited to SGACL policy and IP-SGT & SXP connection can only be externally accessed.
- The below list of trustsec operational data is not supported in Cisco IOS XE Everest 16.5.1:
  - Cisco Trustsec PAC data, environment data and link-level operation data.
  - IPV6 based SGACL policy, IP-SGT mapping and SXP connection operational data.
  - VFR based IP-SGT mapping and SXP connection operational data.

Information About Cisco TrustSec Operational Data

Applications such as YTOOL provides users the flexibility to access Cisco TrustSec operational data from an external interface, without directly logging into Cisco devices to fetch the information using specific commands.

The following types of operational data can be accessed from an external device:

- The active SXP connections on a particular device.

  The following is a sample output to show SXP connections on a device:

```
Device# show cts sxp connections brief
 SXP              : Enabled
 Highest Version Supported: 4
 Default Password : Not Set
 Default Source IP: Not Set
Connection retry open period: 120 secs
Reconcile period: 120 secs
Retry open timer is running
Peer-Sequence traverse limit for export: Not Set
Peer-Sequence traverse limit for import: Not Set

---------------------------------------------------------------------------------------------------
---------------
Peer_IP          Source_IP        Conn Status
 Duration
---------------------------------------------------------------------------------------------------
---------------
10.10.1.1        11.11.1.1        Off
 0:00:36:24 (dd:hr:mm:sec)
10.10.1.2        11.11.1.2        Off
 0:00:36:24 (dd:hr:mm:sec)
10.10.1.3        11.11.1.3        Off
 0:00:36:23 (dd:hr:mm:sec)
10.10.1.4        11.11.1.4        Off
 0:00:36:22 (dd:hr:mm:sec)
10.10.1.5        11.11.1.5        Off
 0:00:36:22 (dd:hr:mm:sec)
10.10.1.6        11.11.1.6        Off
 0:00:36:21 (dd:hr:mm:sec)
10.10.1.7        11.11.1.7        Off
 0:00:36:21 (dd:hr:mm:sec)
```

```
10.10.1.8        11.11.1.8        Off
 0:00:36:20 (dd:hr:mm:sec)
10.10.1.9        11.11.1.9        Off
 0:00:36:15 (dd:hr:mm:sec)
10.10.1.10       11.11.1.10       Off(Speaker)::Off(Listener)
 0:00:33:40 (dd:hr:mm:sec)::0:00:33:40 (
dd:hr:mm:sec)
```

- The IP-SGT mapping information.

  Every source IP is mapped with the corresponding SGT and an IP-SGT binding is created. This mapping information is stored in the Role-Based Manager (RBM) database.

  The following is a sample output to show IP-SGT mapping information:

```
Device# show cts role-based sgt-map all
Active IPv4-SGT Bindings Information

IP Address              SGT     Source
============================================
10.10.10.10             10      CLI
20.20.20.20             20      CLI
30.30.30.30             30      CLI
32.1.1.32               40      CLI
45.1.1.45               100     CLI
69.1.1.1                103     CLI

IP-SGT Active Bindings Summary
============================================
Total number of CLI      bindings = 6
Total number of active   bindings = 6

asr1k-cts-2006#
```

- Names of the policies that are currently applied for every data path.

  SGACL policies are enforced when SGT-tagged packets are transported between two trustsec-aware end points. A policy can either be static or dynamic. Policies that are configured on the device using the CLI command **cts role-based permissions** are static policies. Dynamic policies are configured on CISCO ISE (Identiy Services Engine). Dynamic policies take precedence over static policies. A static policy is enforced only in the absence of a dynamic policy.

  The following is a sample output to show policies for SGT-tagged traffic:

```
Device# show cts role-based permissions
IPv4 Role-based permissions default:
        Permit IP-00
IPv4 Role-based permissions from group 10:SGT_10 to group 10:SGT_10:
        Collab1-10
IPv4 Role-based permissions from group 10:SGT_10 to group 20:SGT_20:
        SGACL_2-30
IPv4 Role-based permissions from group 11:SGT_11 to group 20:SGT_20:
        SGACL_2-30
        SGACL_3-10
        SGACL_4-90
IPv4 Role-based permissions from group 12:SGT_12 to group 20:SGT_20:
        SGACL_3-10
IPv4 Role-based permissions from group 13:SGT_13 to group 20:SGT_20:
        SGACL_4-90
IPv4 Role-based permissions from group 14:SGT_14 to group 20:SGT_20:
        SGACL_5-20
IPv4 Role-based permissions from group 15:SGT_15 to group 20:SGT_20:
        SGACL_6-30
```

```
IPv4 Role-based permissions from group 16:SGT_16 to group 20:SGT_20:
        SGACL_101-90
IPv4 Role-based permissions from group 17:SGT_17 to group 20:SGT_20:
        SGACL_2-30
IPv4 Role-based permissions from group 18:SGT_18 to group 20:SGT_20:
        SGACL_3-10
IPv4 Role-based permissions from group 19:SGT_19 to group 20:SGT_20:
        SGACL_3-10
IPv4 Role-based permissions from group 10:SGT_10 to group 30:SGT_30:
        SGACL_6-30
IPv4 Role-based permissions from group 10:SGT_10 to group 40:SGT_40:
        SGACL_2-30
IPv4 Role-based permissions from group 10:SGT_10 to group 100:SGT_100:
        SGACL_4-90
IPv4 Role-based permissions from group 102:SGT_102 to group 100:SGT_100:
        Permit IP-00
IPv4 Role-based permissions from group 102:SGT_102 to group 103:SGT_103:
        SGACL_2-30
RBACL Monitor All for Dynamic Policies : FALSE
RBACL Monitor All for Configured Policies : FALSE

asr1k-cts-2006#
```

- The contents of each policy - which includes the ACEs (Access Control Entries) in the policy, and the lifetime and refresh time of the policy.

  A policy can have upto a combination of 256 ACEs. Lifetime and refresh time information is only applicable to dynamic policies. The lifetime and refresh time value for a static policy is 0.

  The following is a sample output to show policies for SGT-tagged traffic (only a part of the output is displayed):

```
Device# show cts policy sgt
CTS SGT Policy
==============
RBACL Monitor All : FALSE
RBACL IP Version Supported: IPv4
SGT: 0-02:Unknown
SGT Policy Flag: 0x41408001
RBACL Source List: Empty
RBACL Destination List: Not exist
RBACL Multicast List: Not exist
RBACL Policy Lifetime = 1800 secs
RBACL Policy Last update time = 15:56:42 IST Mon Feb 20 2017
Policy expires in 0:00:03:04 (dd:hr:mm:sec)
Policy refreshes in 0:00:03:04 (dd:hr:mm:sec)
Cache data applied = NONE

SGT: 65535-52:ANY
SGT Policy Flag: 0x41400001
RBACL Source List:
  Source SGT: 65535-52:ANY-0, Destination SGT: 65535-52:ANY-0
  rbacl_type = 80
  rbacl_index = 1
  name   = Permit IP-00
  IP protocol version = IPV4
  refcnt = 4
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
    permit ip
```

```
RBACL Destination List: Not exist
RBACL Multicast List: Not exist
RBACL Policy Lifetime = 1800 secs
RBACL Policy Last update time = 15:56:43 IST Mon Feb 20 2017
Policy expires in 0:00:03:05 (dd:hr:mm:sec)
Policy refreshes in 0:00:03:05 (dd:hr:mm:sec)
Cache data applied = NONE

SGT: 10-2770:SGT_10
SGT Policy Flag: 0x41400001
RBACL Source List:
  Source SGT: 10-2770:SGT_10-0, Destination SGT: 10-2770:SGT_10-0
  rbacl_type = 80
  rbacl_index = 1
  name   = Collab1-10
  IP protocol version = IPV4
  refcnt = 2
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
    permit ip

RBACL Destination List: Not exist
RBACL Multicast List: Not exist
RBACL Policy Lifetime = 1800 secs
RBACL Policy Last update time = 15:56:43 IST Mon Feb 20 2017
Policy expires in 0:00:03:04 (dd:hr:mm:sec)
Policy refreshes in 0:00:03:04 (dd:hr:mm:sec)
Cache data applied = NONE

SGT: 20-44:SGT_20
SGT Policy Flag: 0x41400001
RBACL Source List:
  Source SGT: 10-2770:SGT_10-0, Destination SGT: 20-44:SGT_20-0
  rbacl_type = 80
  rbacl_index = 1
  name   = SGACL_2-30
  IP protocol version = IPV4
  refcnt = 8
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
    permit ip

  Source SGT: 12-17:SGT_12-0, Destination SGT: 20-44:SGT_20-0
  rbacl_type = 80
  rbacl_index = 2
  name   = SGACL_3-10
  IP protocol version = IPV4
  refcnt = 5
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
    permit ip

  Source SGT: 13-14:SGT_13-0, Destination SGT: 20-44:SGT_20-0
  rbacl_type = 80
  rbacl_index = 3
  name   = SGACL_4-90
  IP protocol version = IPV4
  refcnt = 5
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
```

```
  deny tcp

Source SGT: 14-14:SGT_14-0, Destination SGT: 20-44:SGT_20-0
rbacl_type = 80
rbacl_index = 4
name   = SGACL_5-20
IP protocol version = IPV4
refcnt = 2
flag   = 0x41000000
stale  = FALSE
RBACL ACEs:
  permit ip

Source SGT: 15-1410:SGT_15-0, Destination SGT: 20-44:SGT_20-0
rbacl_type = 80
rbacl_index = 5
name   = SGACL_6-30
IP protocol version = IPV4
refcnt = 4
flag   = 0x41000000
stale  = FALSE
RBACL ACEs:
  permit icmp log
  permit udp log
  permit tcp log

Source SGT: 16-14:SGT_16-0, Destination SGT: 20-44:SGT_20-0
rbacl_type = 80
rbacl_index = 6
name   = SGACL_101-90
IP protocol version = IPV4
refcnt = 2
flag   = 0x41000000
stale  = FALSE
RBACL ACEs:
  permit ip
```

How to Configure the External Device YTOOL

Before you configure the YTOOL, ensure that the NETCONF or RESTCONF protocol is enabled on the Cisco device. One of these protocols is required for the YTOOL to communicate with the Cisco device.

Note To enable the NETCONF protocol, use the command **netconf-yang** in the configuration mode. After enabling NETCONF, execute the CLI **show onep session all** to check if the three processes that are needed to use Netconf are running. Netconf is usable only after these three processes are running.

Also, identify the IP address that you are going to use for communicating with the device.

Note YTOOL is also known as yang-explorer. You can download this application from the following location:

Yang Explorer at

To connect the YTOOL to a Cisco device, add the Cisco device in the YTOOL. Steps to add a Cisco device in the YTOOL:

1. Open YTOOL
2. Select **Admin**
3. On the **Ytool Utilities** page, select **Manage Profiles** (under **Manage Device Profiles**)
4. Choose **New Device** from the **Device Profile Name** dropdown
5. On the **Manage Device Profile** page, provide all the details of the device such as **Test Device IP Address, Test Device SSH Port Number, Netconf Username, NetConf Password** etc.

   Figure 22: Manage Device Profile

6. To check the connectivity to the device, navigate to **Build** > **Device Settings**. Select your device from **Profile** and click **Hello**. If you see a response under **Console**, it implies that the YTOOL is able to communicate with the device.

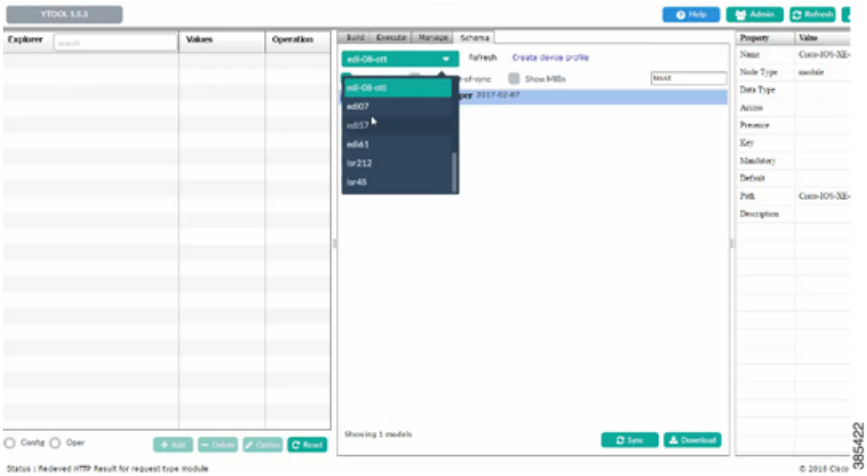
Note To communicate with Cisco devices, you can choose other external applications that can handle data that is structured using YANG. This section is relevant only if you have selected YTOOL to access Cisco devices.

Accessing Operational Data

Before you begin, ensure that the Cisco device from which you are going to extract operational data is configured on the YTOOL. See the "How to Configure the External Device YTOOL" section for details.

1. Download the Cisco TrustSec operational information schema from the Cisco device:
   a. Select **Schema**.
   b. Select the device. The list of schemas in the device will be displayed.

Figure 23: Select a Device

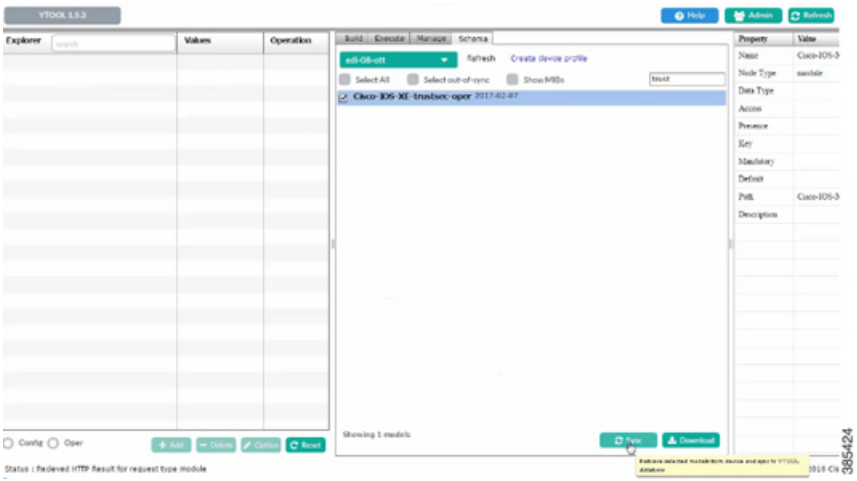


c. Select the Cisco TrustSec operational information schema. Use the search box to search for this schema.

Note The name of an operational information schema ends with **oper**.

d. Click **Sync**. The schema is downloaded into the YTOOL.

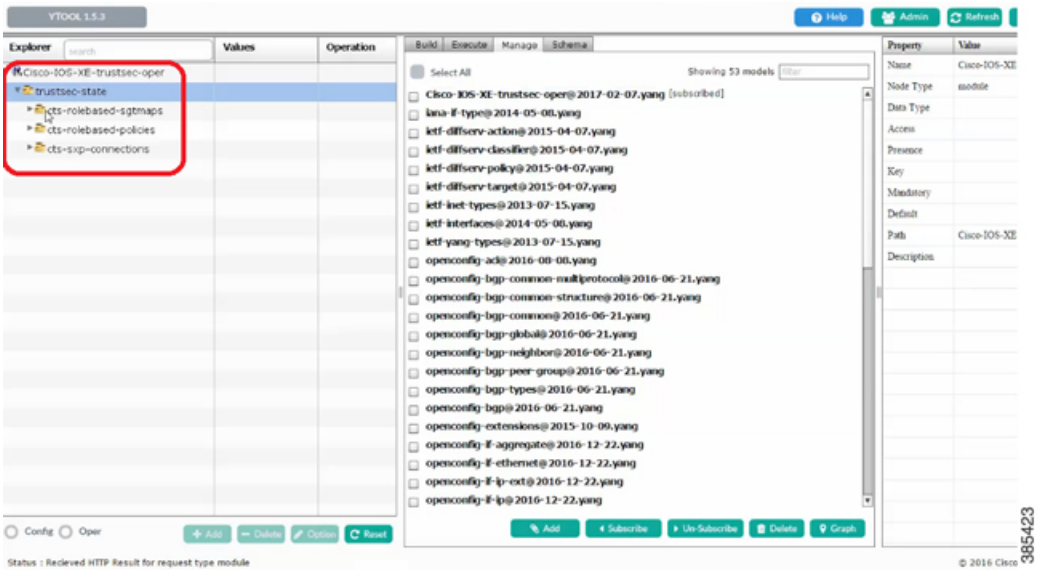
Figure 24: Download Schema



2. Subscribe to the downloaded operational information schema on YTOOL.
   a. Select **Manage**.
   b. From the list of schemas, select the operational information schema.
   c. Click **Subscribe**.

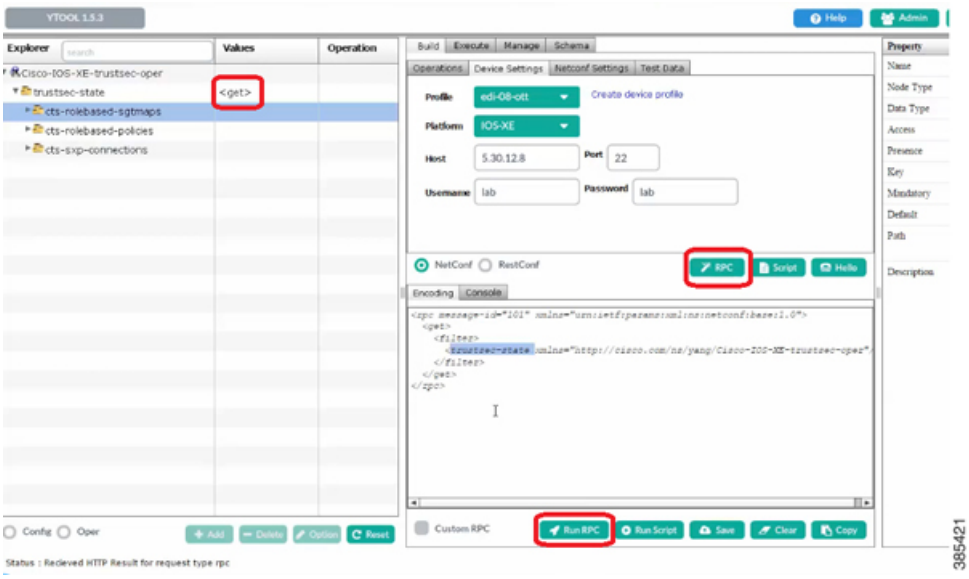
Note Once you have subscribed, the schema will be displayed under explorer.

Figure 25: Subscribe Schema



3. Retrieve selected operational data using the schema:
   a. Against the relevant information level of the operation information schema, select **get** under **values** .
   b. Click **RPC**. An XML generated RPC message will be generated.
   c. Click **Run RPC**. The operation data is retrieved from the Cisco device in the RPC-generated XML format.

Figure 26: Retrieve Operational Data

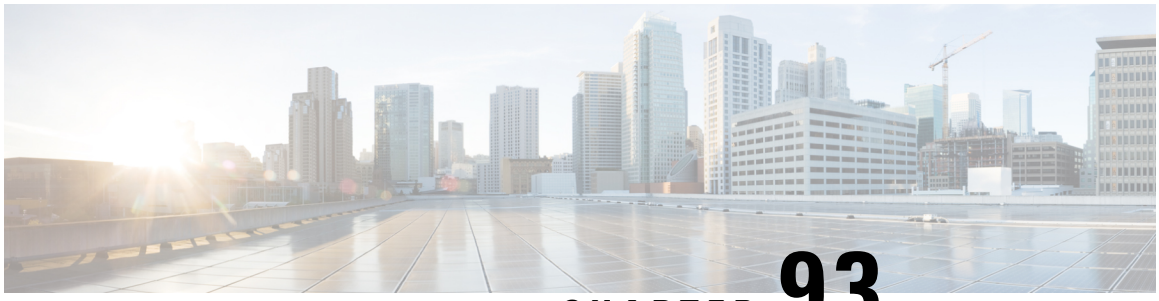


Note For information on the commands that are used to access operational data, see the section Information About Cisco TrustSec Operational Data, on page 964.

Note To communicate with Cisco devices, you can choose other external applications that can handle data that is structured using YANG. This section is relevant only if you have selected YTOOL to access Cisco devices.

PART VIII

Access Node Control Protocol



CHAPTER 93

Access Node Control Protocol

The Access Node Control Protocol (ANCP) feature enhances communication between Digital Subscriber Line Access Multiplexers (DSLAMs) and a broadband remote access server (BRAS), enabling the exchange of events, actions, and information requests between the multiplexer end and the server end. As a result, either end can implement appropriate actions.

Prerequisites for Access Node Control Protocol

To run ANCP over Transmission Control Protocol (TCP), IP must be enabled on broadband remote access servers (BRAS). Interactions from RADIUS to the BRAS are not required for ANCP and are dependent on the RADIUS server.

For information about release and platform support, see the Feature Information for Access Node Control Protocol, on page 987.

Restrictions for Access Node Control Protocol

Cisco IOS XE Release 2.4 supports interactions with the RADIUS server from the broadband remote access server (BRAS). Interactions from RADIUS to the BRAS are not required for ANCP and are dependent on the RADIUS server.

Information About Access Node Control Protocol

ANCP is used to aggregate traffic from multiple subscribers and deliver information for any application, while remaining independent from the application. ANCP is currently used in the application between DSLAMs and the broadband remote access server in a digital subscriber line (DSL) broadband environment.

The ANCP feature enables close communication between DSL aggregation multiplexers (DSLAMs) and network edge devices. Using ANCP between DSLAMs and a BRAS enables exchange of events, actions, and information requests so that the appropriate actions occur at the DSLAM and BRAS.

The ANCP architecture supports the following uses of ANCP:

Rate Adaptive Mode

Rate adaptive mode helps to maximize the line bit rate for a given line, and the rate is dependent on the quality of the signal achieved on the line. Rate adaptive mode conveys DSL modem line rate from a DSLAM to a broadband remote access server.

A BRAS running ANCP listens for TCP requests from its ANCP neighbors (DSLAMs).

- After a TCP session is established--ANCP begins exchanging messages to establish adjacency between the BRAS and its neighbors.
- After adjacency is established--ANCP event messages can be sent from the DSLAM to the BRAS.

Rate adaptive DSL uses signal quality to adjust line speeds. A BRAS typically sets the subscriber interfaces to the maximum bandwidth agreed to in the service license agreement (SLA).

When customer premises equipment (CPE) is synchronized to a data rate that is lower than the line speed, cell or packet loss occurs on the DSLAM. To prevent this, the DSLAM can use ANCP to notify the BRAS of newly adjusted circuit rates.

When a customer-facing port:

- Activates -- The DSLAM sends a Port Up message to the BRAS. The appropriate quality of service (QoS) takes effect in accordance with the ANCP-delivered information.
- Deactivates -- The DSLAM sends a Port Down message to the BRAS. ANCP reports the DSL state sent by the DSLAM, which is typically Silent or Idle. If the broadband remote access server receives another Port Up message, the subscriber sessions either time out or are renewed with a new shaping rate. The shaping rate on the interface does not change until the router receives a new Port Up message.

RADIUS Interaction

Interactions between the broadband remote access server and the RADIUS server are from the router to RADIUS.

The BRAS sends the following attributes and attribute-value pairs (AVPs) to the RADIUS server:

ANCP Line Rates	Upstream Data Rate	Downstream Data Rate	Output Policy Name
VSA 39	Attribute 197, Ascend-Data-Rate	Attribute 255, Ascend-Xmit-Rate	Attribute 77, Connect-Speed-Info
	Attribute Type 38, Rx Connect Speed AVP	Attribute Type 24, Tx Connect Speed AVP	

The BRAS uses Point-to-Point Protocol (PPPoE) to interact with the authentication, authorization, and accounting (AAA) module. RADIUS processes the information and then takes appropriate action.

Port Mapping

Port mapping associates customer premises equipment (CPE) clients of a DSLAM with VLAN subinterfaces on the BRAS. The VLANs include 802.1Q or queue-in-queue (Q-in-Q) hierarchical VLANs. Port mapping is configured in global configuration mode on the BRAS by grouping CPE client IDs with a specific DSLAM neighbor.

There are two methods you can use to map ports: configure all VLAN subinterfaces first, and the ANCP neighbor mappings next. Or, you can configure the mappings directly under the interface.

For example, the following commands configure port mapping for Q-in-Q VLAN subinterfaces:

```
ancp neighbor name
dslam-name
id
dslam-id
dot1q

outer-vlanid
 second-dot1q

inner-vlanid
 [interface

type number
] client-id
 "
client-id
"
```

or

```
ancp neighbor name
dslam-name
id
 dslam-id
dot1q

outer-vlanid
 client-id
 "
client-id
"
```

The *client-id*is a unique access-loop-circuit-id that the DSLAM sends to the BRAS for each unique port. The DSLAM sends this ID in the ANCP Port Up event message. The access-loop-circuit-id uses a defined format consisting of an access node identifier and digital subscriber line (DSL) information as mentioned below:

ATM/DSL

" *access-node-identifier* **atm** *slot/module/port* **.** *subinterface* **:** *vpi* **.** *vci* "

Ethernet/DSL

" *access-node-identifier* **ethernet** *slot* **/** *module* **/** *port* **.** *subinterface* [**:** *vlan-id*]"

The BRAS sets the default state as Down, on all ports of the router, until the DSLAM sends a Port Up message.

Noninteractive Operation Administration and Maintenance

ANCP provides an out-of-band control channel for performing noninteractive operation, administration, and maintenance (OAM) operations from the broadband remote access server. This channel enables router operators to view the ANCP port state of specific DSLAM ports. ANCP port state information is stored in the ANCP dynamic database on the BRAS.

Interactive OAM

The Interactive OAM and Scaling Improvements feature adds on-demand ping capability to ANCP for operations and troubleshooting.

Note This feature is enabled by default and requires no configuration.

General Switch Management Protocol and ANCP

ANCP is an extension of the General Switch Management Protocol (GSMP). GSMP defines a primary-secondary neighbor relationship in which the primary neighbor initiates a connection to a secondary neighbor. In ANCP, this primary-secondary relationship is reversed: the BRAS (primary) listens and accepts incoming ANCP connections from the DSLAM (secondary). The DSLAM uses event messages to communicate asynchronous events to the BRAS, such as topology changes and Port Down or Port Up events.

GSMP connectivity between the BRAS and the DSLAM occurs over TCP/IP (RFC 3293). The DSLAM initiates the connection to the router and the router accepts the connection if the appropriate interface is ANCP enabled.

The GSMP Adjacency Protocol establishes GSMP neighbor relationships.

1. During the adjacency-building:
   a. The DSLAM and router negotiate their capabilities and determine the synchronization state between the two ends.
   b. GSMP detects whether the router and the DSLAM have retained a local information database state in case of a transport failure, or whether both devices require a state update.
   c. If GSMP determines that it must resynchronize the adjacency, it restarts the adjacency synchronization process, which includes the capability negotiation defined in the ANCP extension draft available at:

http://tools.ietf.org/id/draft-wadhwa-gsmp-l2control-configuration-02.txt

1. In an ANCP, if a neighbor (neighbor1) contains capabilities that its neighbor (neighbor2) does not support, neighbor1 turns off the capabilities and recommunicates the packets to neighbor2 with the same set of capabilities as neighbor2.
2. After both the neighbors agree to the same set of capabilities, adjacency is established.

How to Configure Access Node Control Protocol

To configure ANCP, perform the following global or interface configuration tasks:

Enabling ANCP on an Ethernet Interface

Perform this task to enable ANCP on an Ethernet interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ancp adjacency timer** *interval*
4. **interface** *type number*
5. **ip address** *address mask*
6. **ancp enable**
7. **interface** *type number* **.** *subinterface*
8. **encapsulation dot1q** *vlanid* [**second-dot1q** *second-vlanid*]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ancp adjacency timer** *interval* **Example:** `Router(config)# ancp adjacency timer 100`	Sets the ANCP adjacency timer interval, which specifies the amount of time to wait before sending an ANCP hello packet to the DSLAM.
Step 4	**interface** *type number* **Example:** `Router(config)# interface FastEthernet1/0/0`	Enters interface configuration mode to define an interface.
Step 5	**ip address** *address mask* **Example:**	Assigns an IP address and subnet mask to the interface.

	Command or Action	Purpose
	`Router(config-if)# ip address 10.16.1.2 255.255.0.0`	
Step 6	**ancp enable** **Example:** `Router(config-if)# ancp enable`	Enables ANCP on the interface where IP is configured.
Step 7	**interface** *type number* **.** *subinterface* **Example:** `Router(config-if)# interface FastEthernet1/0/0.1`	Enters subinterface configuration mode to define a subinterface.
Step 8	**encapsulation dot1q** *vlanid* [**second-dot1q** *second-vlanid*] **Example:** `Router(config-subif)# encapsulation dot1q 100 second-dot1q 200`	Enables dot1q VLAN encapsulation on the subinterface for a single-queue 802.1Q VLAN or for Q-in-Q hierarchical VLANs.
Step 9	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode.

Enabling ANCP on an ATM Interface

The **ancp enable** command should be configured only for the control VCs on which the ANCP message is sent from the DSLAM. Perform this task to enable ANCP on ATM interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ancp adjacency timer** *interval*
4. **interface atm** *slot* **/** *subslot* **/** *port* **.** *subinterface*
5. **ip address** *ip-address mask*
6. **pvc** *vpi* **/** *vci*
7. **ancp enable**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ancp adjacency timer** *interval* **Example:** `Router(config)# ancp adjacency timer 100`	Sets the ANCP adjacency timer interval, which specifies the amount of time to wait before sending an ANCP hello packet to the DSLAM.
Step 4	**interface atm** *slot* **/** *subslot* **/** *port* **.** *subinterface* **Example:** `Router(config)# interface atm 2/0/1.1`	Enters subinterface configuration mode to define a subinterface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-subif)# ip address 10.16.1.2 255.255.0.0`	Assigns an IP address and subnet mask to the subinterface.
Step 6	**pvc** *vpi* **/** *vci* **Example:** `Router(config-subif)# pvc 2/100`	Enters ATM virtual circuit configuration mode to enable an ANCP connection over ATM PVC.
Step 7	**ancp enable** **Example:** `Router(config-if-atm-vc)# ancp enable`	Enables ANCP on the interface where IP is configured.
Step 8	**exit** **Example:** `Router(config-if-atm-vc)# exit`	Exits ATM virtual circuit configuration mode.

Mapping DSLAM Ports to VLAN Interfaces on Broadband Remote Access Servers

Perform this task to map DSLAM ports to VLAN interfaces on the BRAS.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ancp atm shaper percent-factor** *factor*
4. **interface** *type number.subinterface*
5. **encapsulation dot1q** *vlan-id*
6. **ancp neighbor name** *dslam-name* [**id** *dslam-id*] **client-id** *client-id*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ancp atm shaper percent-factor** *factor* **Example:** `Router(config)# ancp shaper percent-factor 95`	Enables ANCP cell tax accounting for ATM U-interface connections
Step 4	**interface** *type number.subinterface* **Example:** `Router(config)# interface FastEthernet0/0.1`	Enters interface configuration mode for the specified subinterface.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 411`	Enables IEEE 802.1Q encapsulation of traffic on a specified VLAN.
Step 6	**ancp neighbor name** *dslam-name* [**id** *dslam-id*] **client-id** *client-id* **Example:** `Router(config-subif)# ancp neighbor name dslam1 id 1.2.3.4 client-id "1.2.3.4. eth 0/0.1"`	Specifies the ANCP access DSLAM to which VLAN subinterfaces are mapped.
Step 7	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode.

Mapping DSLAM Ports to PVC Interfaces on Broadband Remote Access Servers

The **ancp neighbor name** command is available under **pvc** and **pvc-in-range** command modes. This command creates a one-to-one mapping between a PVC and a DSLAM port. Perform this task to map DSLAM ports to PVC interfaces on the BRAS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ancp atm shaper percent-factor** *factor*
4. **interface atm** *slot* / *subslot* / *port* **.** *subinterface*
5. Do one of the following:
   - **pvc** *vpi* / *vci*
   -
   - **range pvc** *start-vpi* / *start-vci* *end-vpi* / *end-vci*
6. **pvc-in-range** *vpi* / *vci*
7. **ancp neighbor name** *dslam-name* [**id** *dslam-id*] **client-id** *client-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ancp atm shaper percent-factor** *factor* **Example:** `Router(config)# ancp shaper percent-factor 95`	Enables ANCP cell tax accounting for ATM U-interface connections.
Step 4	**interface atm** *slot* / *subslot* / *port* **.** *subinterface* **Example:** `Router(config)# interface atm 2/0/1.1`	Enters interface configuration mode for the specified ATM subinterface.
Step 5	Do one of the following: • **pvc** *vpi* / *vci* •	Creates a one-to-one mapping between a PVC and DSLAM port and enters ATM virtual circuit configuration mode. or

	Command or Action	Purpose

	• **range pvc** *start-vpi* / *start-vci end-vpi* / *end-vci*
	Example:
	`Router(config-subif)# pvc 1/101`
	Example:
	Example:
	`Router(config-subif)# range pvc 9/100 9/102`
		Defines a range of ATM PVCs and enters PVC range configuration mode.
		• If a range of ATM PVCs are defined, use the **pvc-in-range** command to configure an individual PVC.

Step 6	**pvc-in-range** *vpi* / *vci* **Example:** `Router(config-if-atm-range-pvc)# pvc-in-range 9/100`	(Optional) Configures an individual PVC within a range in PVC range configuration mode.
Step 7	**ancp neighbor name** *dslam-name* [**id** *dslam-id*] **client-id** *client-id* **Example:** `Router(config-if-atm-range-pvc)# ancp neighbor name dslam1 id 1.2.3.4 client-id "1.2.3.4. atm0/0.1"`	Specifies the ANCP access DSLAM to which PVC subinterfaces are mapped. • This command is available under PVC range and ATM virtual circuit configuration modes.
Step 8	**end** **Example:** `Router(config-if-atm-range-pvc)# end`	Exits PVC range configuration mode.

Configuration Examples for Access Node Control Protocol

Enabling Access Node Control Protocol on Ethernet Interfaces Example

The following example shows how to enable ANCP on Ethernet subinterface 2/0/1.

```
interface GigabitEthernet 2/0/1
 ip address 192.168.64.16 255.255.255.0
 ancp enable
!
interface GigabitEthernet 2/0/1.1
 encapsulation dot1q 100 second-dot1q 200
!
ancp adjacency timer 100
```

Enabling Access Node Control Protocol on ATM Interfaces Example

The following example shows how to enable ANCP on ATM subinterface 2/0/1.1.

```
interface ATM2/0/0.1 point-to-point
 description ANCP Link to one DSLAM
 no ip mroute-cache
 ip address 192.168.0.2 255.255.255.252
 pvc 254/32
   protocol ip 192.168.0.1
   ancp enable
   no snmp trap link-status
```

Mapping DSLAM Ports to VLAN Interfaces on the BRAS Example

The following example shows how to map the CPE client ports of a DSLAM to Q-in-Q VLAN subinterfaces on the BRAS. In the example, the DSLAM neighbor named dslam1 with an IP address of 192.68.10.5 has a CPE client port mapped to Q-in-Q VLANs 100 and 200 configured on Ethernet interface 1/0/0.2. Another CPE client port is mapped to Q-in-Q VLANs 100 and 100 configured on Ethernet interface 1/0/0.1.

```
interface GigabitEthernet1/0/0.1
 encapsulation dot1q 100 second-dot1q 100
 ancp neighbor name dslam1 id 192.168.10.5 client-id "192.168.10.5 ethernet1/0/0.2"
!
interface GigabitEthernet1/0/0.2
 encapsulation dot1q 100 second-dot1q 200
 ancp neighbor name dslam1 id 192.168.10.5 client-id "192.168.10.5 ethernet1/0/0.1"
!
ancp atm shaper percent-factor 95
!
```

The example shown above maps the ports directly at the subinterface level. You can also configure all VLAN subinterfaces first, and perform the mappings under ANCP neighbor next, as shown in the following example:

```
interface GigabitEthernet1/0/0.1
 encapsulation dot1q 100 second-dot1q 100
!
interface GigabitEthernet1/0/0.2
 encapsulation dot1q 100 second-dot1q 200
!
ancp atm shaper percent-factor 95
!
ancp neighbor name dslam1 id 192.168.10.5
 dot1q 100 second-dot1q 100 interface GigabitEthernet1/0/0.1 client-id "192.168.10.5
ethernet1/0/0.2"
!
ancp neighbor name dslam1 id 192.168.10.5
 dot1q 100 second-dot1q 200 interface GigabitEthernet1/0/0.2 client-id "192.168.10.5
ethernet1/0/0.2"
```

Mapping DSLAM Ports to PVC Interfaces on the BRAS Example

The **ancp neighbor name** command maps the CPE client ports of a DSLAM to PVC interfaces on the BRAS. This command can be configured either globally or under PVC/PVC-in-Range mode.

In PVC or PVC-in-Range Configuration Mode

In this example, the router interfaces with one DSLAM which has two ports or clients.

```
interface ATM2/0/0.1 point-to-point
  description ANCP Link to one DSLAM
  no ip mroute-cache
  ip address 192.168.0.2 255.255.255.252
  pvc 254/32
     protocol ip 192.168.0.1 255.255.255.252
     ancp neighbor name dslam1 id 192.168.10.5 client-id "dslam-port-x-identifier"
        no snmp trap link-status
      !
interface ATM1/0/0.1 multipoint
  description TDSL clients - default TDSL 1024
  class-int speed:ubr:1184:160:10
  range pvc 10/41 10/160
    service-policy input SET-PRECEDENCE-0
    service-policy output premium-plus:l2c:25088
    pvc-in-range 10/103
      description TDSL client 16 Mbps with ANCP
      class-vc speed:ubr:17696:1184:05
      ancp neighbor name dslam1 id 192.168.10.5 client-id "dslam-port-x-identifier"
       !
  range pvc 11/41 11/160
    service-policy input SET-PRECEDENCE-0
    service-policy output premium-plus:l2c:25088
    pvc-in-range 11/108
      description TDSL client 16 Mbps with ANCP
      class-vc speed:ubr:17696:1184:05
      ancp neighbor name dslam1 id 192.168.10.5 client-id "dslam-port-y-identifier"
        !
```

In Global Configuration Mode

When the **ancp neighbor** command is configured globally, the PVC information for the ATM interface must also be specified, as shown in the following example:

```
interface ATM1/0/0.1 multipoint
 description TDSL clients - default TDSL 1024
 class-int speed:ubr:1184:160:10
 range pvc 10/41 10/160
   service-policy input SET-PRECEDENCE-0
   service-policy output premium-plus:l2c:25088
   pvc-in-range 10/103
     description TDSL client 16 Mbps with ANCP
     class-vc speed:ubr:17696:1184:05
!
 range pvc 11/41 11/160
   service-policy input SET-PRECEDENCE-0
   service-policy output premium-plus:l2c:25088
   pvc-in-range 11/108
     description TDSL client 16 Mbps with ANCP
     class-vc speed:ubr:17696:1184:05
!
ancp neighbor name dslam1 id 192.168.10.5
 atm 10/103 interface ATM1/0/0.1 client-id "dslam-port-x-identifier"
 atm 11/108 interface ATM1/0/0.1 client-id "dslam-port-y-identifier"
```

Additional References for Access Node Control Protocol

Related Documents

Related Topic	Document Title
ANCP Commands	*Cisco IOS Access Node Control Protocol Command Reference*
IEEE 802.1Q VLAN	Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation
Queue-in-Queue VLAN Tags	IEEE 802.1Q-in-Q VLAN Tag Termination

RFCs

RFC	Title
ANCP extension draft	GSMP Extensions for Access Node Control Mechanism, Internet draft
RFC 3292	*General Switch Management Protocol (GSMP) V3*
RFC 3293	General Switch Management Protocol (GSMP), Packet Encapsulations for Asynchronous Transfer Mode (ATM), Ethernet and Transmission Control Protocol (TCP)

Feature Information for Access Node Control Protocol

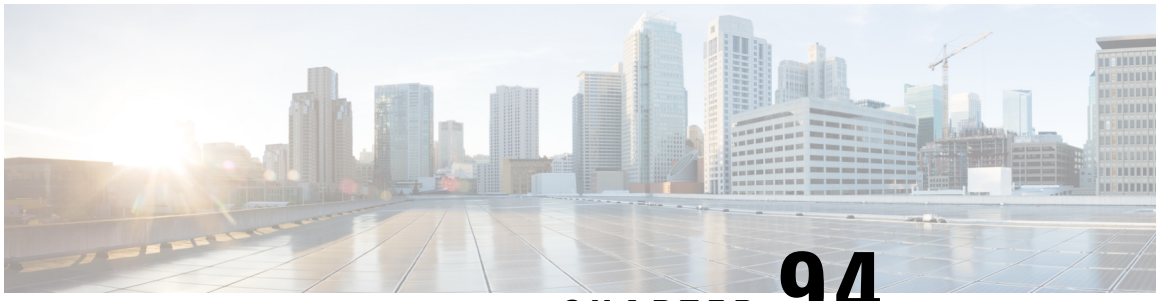
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 128: Feature Information for Access Node Control Protocol

Feature Name	Releases	Feature Information
Access Node Control Protocol	Cisco IOS XE Release 2.4	In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000. The following command was introduced: **ancp vdsl ethernet shaper**.

Feature Name	Releases	Feature Information
Interactive OAM and Scaling Improvements	Cisco IOS XE Release 2.4	The Interactive OAM and Scaling Improvements feature adds on demand ping capability to ANCP for operations and troubleshooting. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000. The following commands were introduced or modified: **ping ancp**, **show ancp neighbor port**, **show ancp port**, **show ancp session**, **show ancp session adjacency**, **show ancp session event**, and **show ancp statistics**.



CHAPTER 94

Multiservice Activation in Access-Accept Message

The Multiservice Activation in Access-Accept Message feature is part of Access Node Control Protocol (ANCP) and allows multiple services to be included in a single RADIUS Access-Accept message.This feature is similar to the Multiservice Activation and Deactivation in a Change of Authorization (CoA) Message feature, but in this case all requested service activations are processed automatically. This means that if a service activation fails, no further service activations are processed, and any service that has already been activated by the Access-Accept message is deactivated.

Restrictions for Multiservice Activation in Access-Accept Message

- If one of the service activations fails, all unprocessed services from the Access-Accept message will be ignored, and any services from the Access-Accept message that have been activated will be deactivated.
- A two-stage application process exists when applying a quality of service (QoS) policy via a service in an Access-Accept message. The first stage involves parsing the policy and sending the policy value to the dataplane. The second stage involves the application of the QoS policy on the dataplane. In the instance where stage one is completed successfully, but stage two fails, the relevant service can indicate that the activation was successful.

Information About Multiservice Activation in Access-Accept Message

Multiservice Activation in Access-Accept Message Overview

An Access-Request message is sent by a RADIUS client to a RADIUS server to authenticate the user or subscriber profile included in the message. If the user or subscriber profile is:

- Acceptable--The RADIUS server may return an Access-Accept message
- Unacceptable--The RADIUS server may return an access-reject message

To enable multiservice activation, the Access-Accept message may include multiple Cisco generic VSA 250 (SSG_ACCOUNT_INFO) entries, with each VSA specifying a service name to be activated.

RSIM Format

```
vsa cisco generic 250 string "Aservice-name1"
vsa cisco generic 250 string "Aservice-name2"
vsa cisco generic 250 string "Aservice-name3"
```

RADIUS Format

```
07:06:23.234: RADIUS: Received from id 1645/36 11.12.13.2:1645, Access-Accept, len 112
07:06:23.238: RADIUS:  authenticator 92 C5 A2 F2 24 56 37 1E - 74 F4 C6 92 B0 E8 92 4C
07:06:23.238: RADIUS:  Vendor, Cisco       [26]  23
07:06:23.238: RADIUS:   ssg-account-info   [250] 17  "Aservice-name-1"
07:06:23.238: RADIUS:  Vendor, Cisco       [26]  23
07:06:23.238: RADIUS:   ssg-account-info   [250] 17  "Aservice-name-2"
07:06:23.238: RADIUS:  Vendor, Cisco       [26]  23
07:06:23.238: RADIUS:   ssg-account-info   [250] 17  "Aservice-name-3"
```

Upon receipt of the Access-Accept message, the specified services are extracted and each service is activated serially. If a service activation fails, all unprocessed services from the Access-Accept message are ignored, and any services from the Access-Accept message that have been activated are deactivated.

Note The RSIM format for Access-Accept multiple services requests for QoS services is not applicable for multiple service activation or deactivation requests in a CoA message. The format for CoA messages is VSA 252. For more information see Multiservice Activation and Deactivation in a CoA Message module

QoS Policy for VSA 250

You can use VSA 250 concatenated QoS syntax with the RADIUS Access-Accept message while establishing a session. The syntax parses the VSA concatenated string and activates the QoS and Intelligent Services Gateway (ISG) policy.

Note ISG manages multiple QoS services in one Access-Accept message and applies the message to activate static and parameterized QoS.

How to Configure Multiservice Activation in Access-Accept Message

Activating a Session Service Using Access-Accept

Configure Cisco VSA 250 in the service profile on RADIUS to dynamically activate a session service with Access-Accept. RADIUS uses VSA 250 in Access-Accept messages with the following syntax:

RSIM Format

```
vsa cisco generic 250 string
"Aservice-name-1"
```

Configuration Examples for Multiservice in Access-Accept Message

Activating QoS Services Using VSA 250 Example

To activate QoS Services, use the *qos:vc-qos-policy-out* syntax with the RADIUS Access-Accept message. The concatenated string is parsed and the QoS and ISG policy is activated.

The following example defines VSA 250 concatenated string parsing, and the activation of the ISG service and QoS policies:

qos:<qos-attribute-name>=<attribute value>[;qos:<qos-attribute-name>=<attribute value>...]

qos-attribute-name	Displays the QoS attribute name. The accepted attributes for the QoS attribute name in this special concatenated format are: vc-qos-policy-in vc-qos-policy-out vc-weight vc-watermark-min vc-watermark-max
attribute value	Displays the value to be assigned to the QoS attribute. The acceptable range of values are determined by the platform.

If the target session is an ATM VC, the vc-weight, vc-watermark-min, and vc-watermark-max attributes are interpreted.

The following example displays the concatenated QoS syntax for VSA 250:

vsa cisco generic 250 string "Aqos:vc-qos-policy-out=IPOne_out;qos:vc-qos-policy-in=IPOne_in"

Additional References for Multiservice Activation in Access-Accept Message

Related Documents

Related Topic	Document Title
ANCP commands	Cisco IOS Access Node Control Protocol Command Reference
IEEE 802.1Q VLAN	Cisco IOS IEEE 802.1Q Support feature module
Access-Node Control Protocol	Metro Ethernet WAN Services and Architectures (white paper), Access Node Control Protocol
Queue-in-Queue VLAN Tags	IEEE 802.1Q-in-Q VLAN Tag Termination

RFCs

RFC	Title
ANCP extension draft	GSMP Extensions for Access Node Control Mechanism, Internet draft
RFC 3292	*General Switch Management Protocol (GSMP) V3*
RFC 3293	*General Switch Management Protocol (GSMP), Packet Encapsulations for Asynchronous Transfer Mode (ATM), Ethernet and Transmission Control Protocol (TCP)*

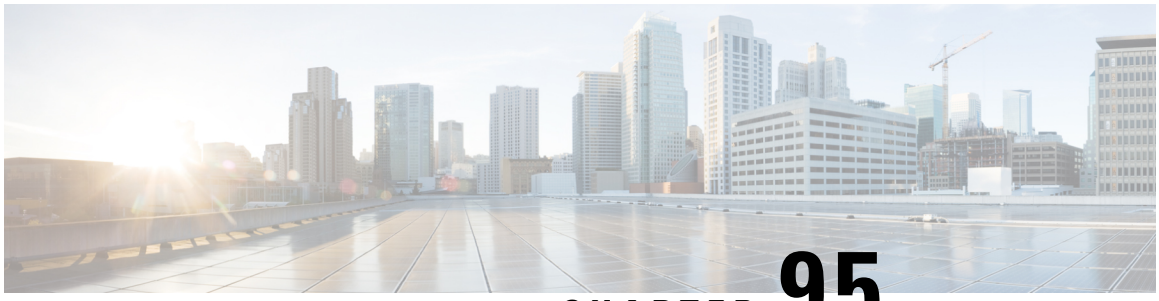
Feature Information for Multiservice Activation in Access-Accept Message

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 129: Feature Information for Multiservice Activation in Access-Accept Message

Feature Name	Releases	Feature Information
Multiservice Activation in Access-Accept Message	Cisco IOS XE Release 2.4	The Multiservice Activation in Access-Accept Message feature supports dynamic activation of multiple services using RADIUS Access-Accept messages. In Cisco IOS XE 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers. The following command was modified by this feature: **subscriber service multiple-accept**.



CHAPTER 95

Multiservice Activation and Deactivation in a CoA Message

This feature allows multiple services to be activated or deactivated by a single Change of Authorization (CoA) message sent from the policy server. This feature is similar to the Multiservice Activation in Access-Accept Message feature, but in this case it is assumed that the user session is already active.

Restrictions for Multiservice Activation and Deactivation in a CoA Message

- All service names included in the multiservice activation or deactivation message must be Intelligent Services Gateway (ISG) aware. For example, they must be of type class-map type service "service1."
- If one of the services activation or deactivation messages fails, the broadband remote access server (BRAS) rolls back only the previous successfully activated or deactivated services and those that were included in the same multiservice activation or deactivation CoA message.
- However, the current ISG implementation has limitations in the process of reestablishing the state of previously activated or deactivated services. For example, if a feature that can overlap is enabled in the same session, the new, successfully activated or deactivated feature parameters delete the old parameters of the same feature, which was already activated in that session. Attempts to reestablish old parameters of that feature fail.
- If a valid CLI-configured ISG service is forwarded through CoA to a new session and fails (ISG service is unable to find an accounting list):
  - BRAS does not wait for the hardware to be provisioned.
  - An ACK message is relayed.
  - ISG services are not applied.
  - Tracebacks are observed.

Information About Multiservice Activation and Deactivation in a CoA Message

Multiservice Activation and Deactivation in a CoA Message Overview

The CoA multiservice activation or deactivation message contains a list of services. Multiple services are listed in the form of multiple lines in a VSA 252.

For the case of multiservice deactivation within one CoA message, the RADIUS server sends the request to deactivate multiple services within one CoA multiservice deactivation message. For each service listed in the multiservice deactivation message, the BRAS deactivates the service. Successful deactivation of the service is followed by an accounting-stop message.

If a service cannot be successfully deactivated, the BRAS terminates the deactivation of all subsequent services contained in the multiservice activation message. The BRAS activates all the services within the same multiservice activation message that were successfully deactivated before the failed service activated.

An existing VSA 252 is used to form one multiservice activation or deactivation CoA message. To form one multiservice activate or deactivate CoA message, multiple lines of VSA 252 are included in the message. The following example shows mixed multiservice activation or deactivation in one CoA message:

RADIUS Format

```
ISG#
00:41:15: RADIUS: CoA  received from id 76 10.168.1.6:1700, CoA Request, len 67
00:41:15: CoA: 10.168.1.6 request queued
00:41:15: RADIUS:  authenticator C4 AC 5D 50 6A BE D7 00 - F9 1D FA 38 15 32 25 3A
00:41:15: RADIUS:  Vendor, Cisco       [26]  18
00:41:15: RADIUS:   ssg-account-info   [250] 12  "S151.1.1.2"
00:41:15: RADIUS:  Vendor, Cisco       [26]  17
00:41:15: RADIUS:   ssg-command-code   [252] 11
00:41:15: RADIUS:   0B 70 6F 6C 69 63 65 31           [Service-Log-On service1]
00:41:15: RADIUS:  Vendor, Cisco       [26]  17
00:41:15: RADIUS:   ssg-command-code   [252] 11
00:41:15: RADIUS:   0B 70 6F 6C 69 63 65 32           [Service-Log-On service2]
00:41:15: RADIUS:  Vendor, Cisco       [26]  17
00:41:15: RADIUS:   ssg-command-code   [252] 11
00:41:15: RADIUS:   0C 73 65 72 76 69 63 65 33        [Service-Log-Off service3]
00:41:15: RADIUS:  Vendor, Cisco       [26]  17
00:41:15: RADIUS:   ssg-command-code   [252] 11
00:41:15: RADIUS:   0B 70 6F 6C 69 63 65 34           [Service-Log-On service4]
```

QoS Policy for VSA 252

You can use VSA 252 concatenated quality of service (QoS) syntax in a RADIUS CoA message. The syntax is used to activate or deactivate ISG service and the QoS policy by parsing the VSA 252 concatenated string.

Note ISG manages multiple QoS services in one CoA message and applies the message to activate static and parameterized QoS.

How to Configure Multiservice Activation and Deactivation in a CoA Message

Activating a Session Service Using CoA

Configure Cisco VSA 252 in the service profile on RADIUS to dynamically activate a session service with CoA. RADIUS uses VSA 252 in CoA messages with the following syntax:

```
vsa cisco generic 252 binary 0b suffix
"qos:vc-qos-policy-out=IPOne_out;qos:vc-qos-policy-in=IPOne_in;;"
```

The CoA command in this example performs the following actions:

- Initiates an ISG service "qos:vc-qos-policy-out=IPOne_out;qos:vc-qos-policy-in=IPOne_in;;".
- Replaces the default QoS output child policy on virtual template IPOne_out and installs the IPOne_out policy if there is no default output child policy on the virtual template.
- Replaces the default QoS input child policy on virtual template IPOne_in and installs the IPOne_in policy if there is no default input child policy configured on the virtual template.

Deactivating a Session Service Using CoA

To dynamically activate a session service using CoA and default QoS policy on a virtual template, configure Cisco VSA 252 in the RADIUS service profile. RADIUS uses VSA 252 in CoA messages with the following syntax:

```
vsa cisco generic 252 binary 0c suffix
"qos:vc-qos-policy-out=IPOne_out;qos:vc-qos-policy-in=IPOne_in;;"
```

The CoA command in this example performs the following actions:

- Terminates an ISG service "qos:vc-qos-policy-out=IPOne_out;qos:vc-qos-policy-in=IPOne_in".
- Replaces the QoS output child policy IPOne_out with the default child policy configured on the appropriate virtual template interface.
- Replaces the QoS input child policy IPOne_in with the default child policy configured on the appropriate virtual template interface.

Configuration Examples for Multiservice Activation and Deactivation in a CoA Message

Activating and Deactivating QoS Services Using VSA 252 Example

To activate QoS services, RADIUS adds one or more multiple QoS classes to the parent and child policy in one VSA 252 string and relays the following syntax:

```
CoA VSA 252 0b <new service>
```

In addition to the existing services, the new service should be installed and should not have overlapping classes with the current services.

The following example defines QoS activation and adds the QoS classes in the parameterized QoS service RADIUS form:

```
VSA252 0b q-p-out=IPOne1-isg-acct_service(1)((c-d,voip)1(200000,9216,0,1,0,0)10(9));q-p-in=
 ((c-d,voip)1(200000,9216,0,1,0,0)10(9))
```

To deactivate the second service, RADIUS relays the same VSA 252 string that was used for service activation, replacing "0b" with "0c".

The following example defines QoS deactivation and deletes the QoS classes in the parameterized QoS service RADIUS form:

```
VSA252 0c q-p-out=IPOne1-isg-acct_service(1)((c-d,voip)1(200000,9216,0,1,0,0)10(9));q-p-in=
 ((c-d,voip)1(200000,9216,0,1,0,0)10(9))
```

Additional References for Multiservice Activation and Deactivation in a CoA Message

Related Documents

Related Topic	Document Title
ANCP Commands	*Cisco IOS Access Node Control Protocol Command Reference*
IEEE 802.1Q VLAN	Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation
Queue-in-Queue VLAN Tags	IEEE 802.1Q-in-Q VLAN Tag Termination

RFCs

RFC	Title
ANCP extension draft	GSMP Extensions for Access Node Control Mechanism, Internet draft

RFC	Title
RFC 3292	*General Switch Management Protocol (GSMP) V3*
RFC 3293	*General Switch Management Protocol (GSMP), Packet Encapsulations for Asynchronous Transfer Mode (ATM), Ethernet and Transmission Control Protocol (TCP)*

Feature Information for Multiservice Activation and Deactivation in a CoA Message

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

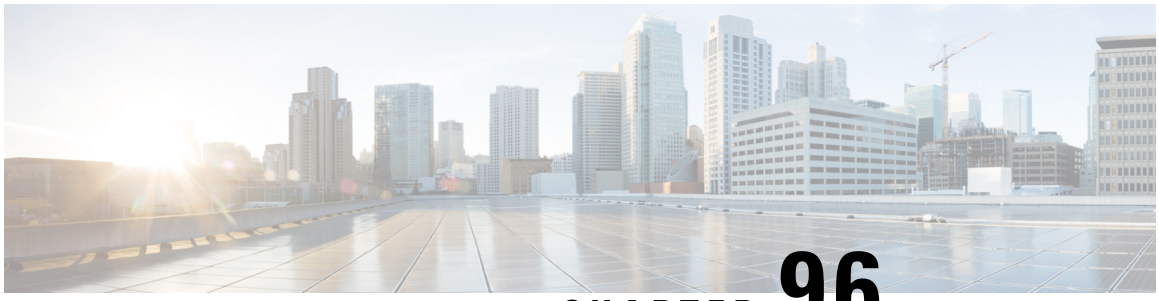
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 130: Feature Information for Multiservice Activation and Deactivation in a CoA Message

Feature Name	Releases	Feature Information
Multiservice Activation and Deactivation in a CoA Message	Cisco IOS XE Release 2.4	The Multiservice Activation and Deactivation in a CoA Message feature supports dynamic activation and deactivation of multiple services using RADIUS CoA messages. In Cisco IOS XE 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers.

PART IX

First Hop Security



CHAPTER 96

IPv6 RA Guard

The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue router advertisement (RA) guard messages that arrive at the network device platform.

Restrictions for IPv6 RA Guard

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.
- This feature can be configured on a switch port interface in the ingress direction.
- This feature supports host mode and router mode.
- This feature is supported only in the ingress direction; it is not supported in the egress direction.
- This feature is not supported on EtherChannel and EtherChannel port members.
- This feature is not supported on trunk ports with merge mode.
- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.
- Packets dropped by the IPv6 RA Guard feature can be spanned.
- If the **platform ipv6 acl icmp optimize neighbor-discovery command** is configured, the IPv6 RA Guard feature cannot be configured and an error message will be displayed. This command adds default global Internet Control Message Protocol (ICMP) entries that will override the RA guard ICMP entries.

Information About IPv6 RA Guard

IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 ND inspection and IPv6 RA guard are IPv6 global policies features. Every time an ND inspection or RA guard is configured globally, the policy attributes are stored in the software policy database. The policy is then applied to an interface, and the software policy database entry is updated to include this interface to which the policy is applied.

IPv6 RA Guard

The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The IPv6 RA Guard feature analyzes these RAs and filters out RAs that are sent by unauthorized devices. In host mode, all RA and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 (L2) device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

In the wireless deployment RAs coming on wireless ports are dropped as routers cannot reside on these interfaces.

How to Configure IPv6 RA Guard

Configuring the IPv6 RA Guard Policy on the Device

Note When the **ipv6 nd raguard** command is configured on ports, router solicitation messages are not replicated to these ports. To replicate router solicitation messages, all ports that face routers must be set to the router role.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd raguard policy** *policy-name*
4. **device-role** {**host** | **router**}
5. **hop-limit** {**maximum** | **minimum** *limit*}
6. **managed-config-flag** {**on** | **off**}
7. **match ipv6 access-list** *ipv6-access-list-name*
8. **match ra prefix-list** *ipv6-prefix-list-name*
9. **other-config-flag** {**on** | **off**}

10. **router-preference maximum** {**high** | **low** | **medium**}
11. **trusted-port**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy policy1`	Defines the RA guard policy name and enters RA guard policy configuration mode.
Step 4	**device-role** {**host**	**router**} **Example:** `Device(config-ra-guard)# device-role router`	Specifies the role of the device attached to the port.
Step 5	**hop-limit** {**maximum**	**minimum** *limit*} **Example:** `Device(config-ra-guard)# hop-limit minimum 3`	(Optional) Enables verification of the advertised hop count limit. • If not configured, this check will be bypassed.
Step 6	**managed-config-flag** {**on**	**off**} **Example:** `Device(config-ra-guard)# managed-config-flag on`	(Optional) Enables verification that the advertised managed address configuration flag is on. • If not configured, this check will be bypassed.
Step 7	**match ipv6 access-list** *ipv6-access-list-name* **Example:** `Device(config-ra-guard)# match ipv6 access-list list1`	(Optional) Enables verification of the sender's IPv6 address in inspected messages from the configured authorized device source access list. • If not configured, this check will be bypassed.
Step 8	**match ra prefix-list** *ipv6-prefix-list-name* **Example:** `Device(config-ra-guard)# match ra prefix-list listname1`	(Optional) Enables verification of the advertised prefixes in inspected messages from the configured authorized prefix list. • If not configured, this check will be bypassed.

	Command or Action	Purpose
Step 9	**other-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# other-config-flag on`	(Optional) Enables verification of the advertised "other" configuration parameter.
Step 10	**router-preference maximum** {**high** | **low** | **medium**} **Example:** `Device(config-ra-guard)# router-preference maximum high`	(Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit.
Step 11	**trusted-port** **Example:** `Device(config-ra-guard)# trusted-port`	(Optional) Specifies that this policy is being applied to trusted ports. • All RA guard policing will be disabled.
Step 12	**exit** **Example:** `Device(config-ra-guard)# exit`	Exits RA guard policy configuration mode and returns to global configuration mode.

Configuring IPv6 RA Guard on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd raguard attach-policy** [*policy-name* [**vlan** {**add** \| **except** \| **none** \| **remove** \| **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]
5. **exit**
6. **show ipv6 nd raguard policy** [*policy-name*]
7. **debug ipv6 snooping raguard** [*filter* \| *interface* \| *vlanid*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 3/13`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd raguard attach-policy** [*policy-name* [**vlan** {**add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]] **Example:** `Device(config-if)# ipv6 nd raguard attach-policy`	Applies the IPv6 RA Guard feature to a specified interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**show ipv6 nd raguard policy** [*policy-name*] **Example:** `Device# show ipv6 nd raguard policy raguard1`	Displays the RA guard policy on all interfaces configured with the RA guard.
Step 7	**debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*] **Example:** `Device# debug ipv6 snooping raguard`	Enables debugging for IPv6 RA guard snooping information.

Configuration Examples for IPv6 RA Guard

Example: IPv6 RA Guard Configuration

```
Device(config)# interface fastethernet 3/13

Device(config-if)# ipv6 nd raguard attach-policy

Device# show running-config interface fastethernet 3/13

Building configuration...
Current configuration : 129 bytes
!
interface FastEthernet3/13
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

Example: Configuring IPv6 ND Inspection and RA Guard

This example provides information about an interface on which both the Neighbor Discovery Inspection and RA Guard features are configured:

```
Device# show ipv6 snooping capture-policy interface ethernet 0/0

Hardware policy registered on Ethernet 0/0
Protocol     Protocol value   Message   Value     Action    Feature
ICMP         58               RS        85        punt      RA Guard
                                                  punt      ND Inspection
ICMP         58               RA        86        drop      RA guard
                                                  punt      ND Inspection
ICMP         58               NS        87        punt      ND Inspection
ICM          58               NA        88        punt      ND Inspection
ICMP         58               REDIR     89        drop      RA Guard
                                                  punt      ND Inspection
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

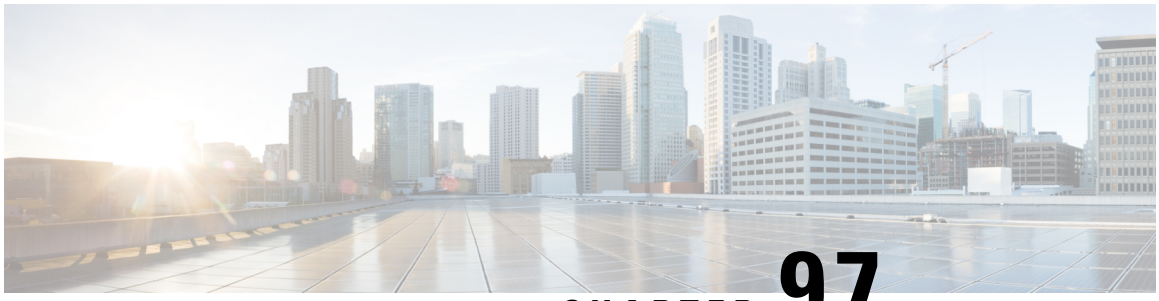
Feature Information for IPv6 RA Guard

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 131: Feature Information for IPv6 RA Guard

Feature Name	Releases	Feature Information
IPv6 RA Guard	12.2(33)SXI4 12.2(50)SY 12.2(54)SG 15.0(2)SE 15.0(2)SG Cisco IOS XE Release 3.8S Cisco IOS XE Release 3.2SE Cisco IOS XE Release 3.2SG	The following commands were introduced or modified: **debug ipv6 snooping raguard**, **device-role**, **hop-limit**, **ipv6 nd raguard attach-policy**, **ipv6 nd raguard policy**, **managed-config-flag**, **match ipv6 access-list**, **match ra prefix-list**, **other-config-flag**, **router-preference maximum**, **show ipv6 nd raguard policy**.



CHAPTER 97

IPv6 Snooping

The IPv6 Snooping feature bundles several Layer 2 IPv6 first-hop security features, including IPv6 neighbor discovery inspection, IPv6 device tracking, IPv6 address glean, and IPv6 binding table recovery, to provide security and scalability. IPv6 ND inspection operates at Layer 2, or between Layer 2 and Layer 3, to provide IPv6 functions with security and scalability.

Restrictions for IPv6 Snooping

The IPv6 snooping feature is not supported on Etherchannel ports.

Information About IPv6 Snooping

The following sections provide information about IPv6 snooping.

IPv6 Snooping

The IPv6 Snooping feature bundles several Layer 2 IPv6 first-hop security features, including IPv6 Address Glean and IPv6 Device Tracking. The feature operates at Layer 2, or between Layer 2 and Layer 3, and provides IPv6 features with security and scalability. This feature mitigates some of the inherent vulnerabilities for the neighbor discovery mechanism, such as attacks on duplicate address detection (DAD), address resolution, device discovery, and the neighbor cache.

IPv6 Snooping learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables and analyzes ND messages in order to build a trusted binding table. IPv6 ND messages that do not have valid bindings are dropped. An ND message is considered trustworthy if its IPv6-to-MAC mapping is verifiable.

When IPv6 Snooping is configured on a target (which varies depending on platform target support and may include device ports, switch ports, Layer 2 interfaces, Layer 3 interfaces, and VLANs), capture instructions are downloaded to the hardware to redirect the ND protocol and Dynamic Host Configuration Protocol (DHCP) for IPv6 traffic up to the switch integrated security features (SISF) infrastructure in the routing device. For

ND traffic, messages such as NS, NA, RS, RA, and REDIRECT are directed to SISF. For DHCP, UDP messages sourced from port 546 or 547 are redirected.

IPv6 Snooping registers its "capture rules" to the classifier, which aggregates all rules from all features on a given target and installs the corresponding ACL down into the platform-dependent modules. Upon receiving redirected traffic, the classifier calls all entry points from any registered feature (for the target on which the traffic is being received), including the IPv6 snooping entry point. This entry point is the last to be called, so any decision (such as drop) made by another feature supersedes the IPv6 Snooping decision.

IPv6 Device Tracking

IPv6 device tracking provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

IPv6 First-Hop Security Binding Table

The IPv6 First-Hop Security Binding Table recovery mechanism feature enables the binding table to recover in the event of a device reboot. A database table of IPv6 neighbors connected to the device is created from information sources such as ND snooping. This database, or binding, table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

This mechanism enables the binding table to recover in the event of a device reboot. The recovery mechanism will block any data traffic sourced from an unknown source; that is, a source not already specified in the binding table and previously learned through ND or DHCP gleaning. This feature recovers the missing binding table entries when the resolution for a destination address fails in the destination guard. When a failure occurs, a binding table entry is recovered by querying the DHCP server or the destination host, depending on the configuration.

Recovery Protocols and Prefix Lists

The IPv6 First-Hop Security Binding Table Recovery Mechanism feature introduces the capability to provide a prefix list that is matched before the recovery is attempted for both DHCP and NDP.

If an address does not match the prefix list associated with the protocol, then the recovery of the binding table entry will not be attempted with that protocol. The prefix list should correspond to the prefixes that are valid for address assignment in the Layer 2 domain using the protocol. The default is that there is no prefix list, in which case the recovery is attempted for all addresses. The command to associate a prefix list to a protocol is **protocol** {**dhcp** | **ndp**} [**prefix-list** *prefix-list-name*].

IPv6 Device Tracking

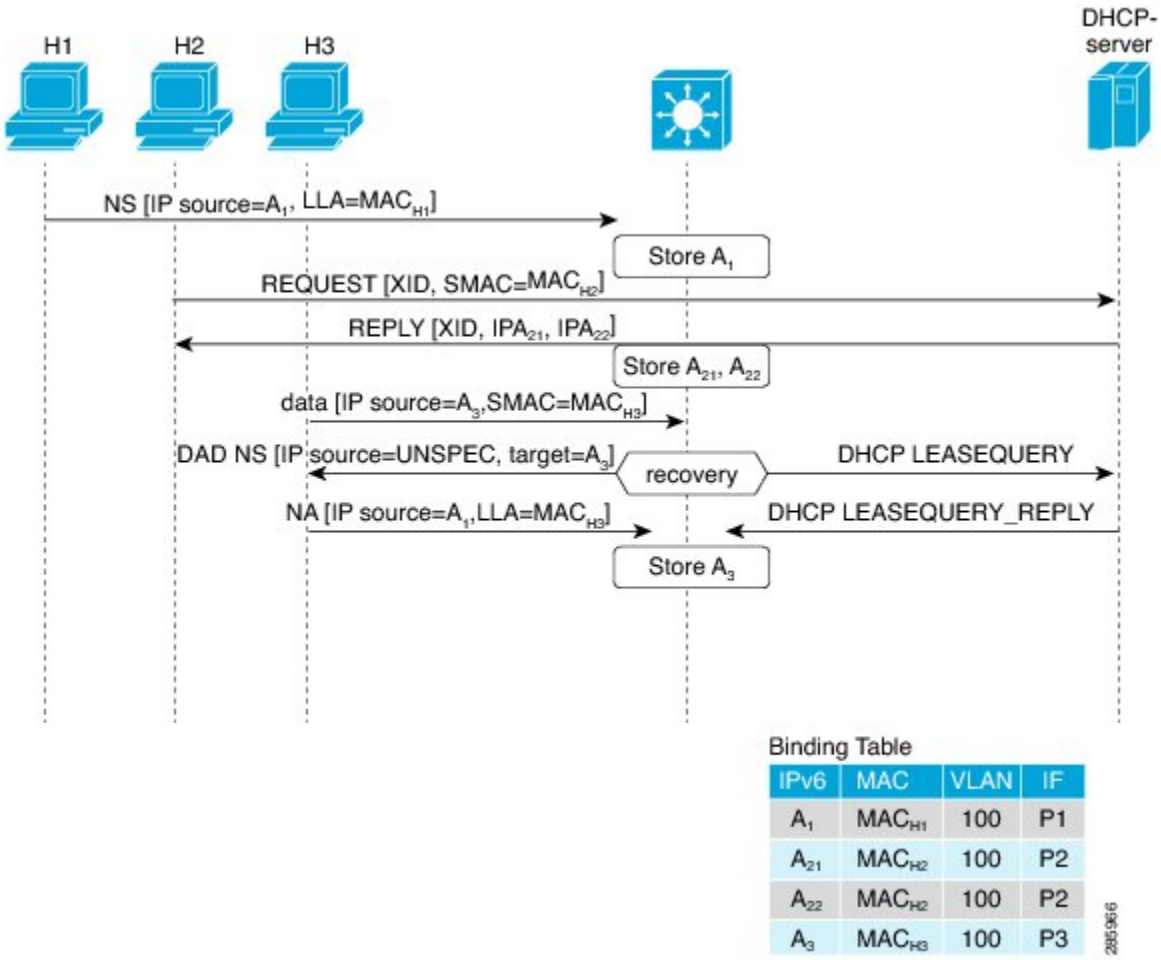
IPv6 device tracking provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

IPv6 Address Glean

IPv6 address glean is the foundation for many other IPv6 features that depend on an accurate binding table. It inspects ND and DHCP messages on a link to glean addresses, and then populates the binding table with these addresses. This feature also enforces address ownership and limits the number of addresses any given node is allowed to claim.

The following figure shows how IPv6 address glean works.

Figure 27: IPv6 Address Glean



Support for Multiple IA_NA and IA_PD

In some cases, a network device can request and receive more than one IPv6 address from the DHCP server. This may be done to provide addresses to multiple clients of the device, such as when a residential gateway requests addresses to distribute to its LAN clients. When the device sends out a DHCPv6 packet, the packet includes all of the addresses that have been assigned to the device.

When SISF analyzes a DHCPv6 packet, it examines the IA_NA (Identity Association-Nontemporary Address) and IA_PD (Identity Association-Prefix Delegation) components of the packet, and extracts each IPv6 address contained in the packet. SISF adds each extracted address to the binding table.

How to Configure IPv6 Snooping

Configuring IPv6 Snooping on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 snooping policy** *snooping-policy*
4. **exit**
5. **interface** *type number*
6. **ipv6 snooping attach-policy** *snooping-policy*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 snooping policy** *snooping-policy* **Example:** `Device(config)# ipv6 snooping policy policy1`	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.
Step 4	**exit** **Example:** `Device(config-ipv6-snooping)# exit`	Exits IPv6 snooping configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 0/0/1`	Enters interface configuration mode.
Step 6	**ipv6 snooping attach-policy** *snooping-policy* **Example:** `Device(config-if)# ipv6 snooping attach-policy policy1`	Attaches the IPv6 snooping policy to the interface.

Verifying and Troubleshooting IPv6 ND Inspection

SUMMARY STEPS

1. **enable**
2. **show ipv6 snooping capture-policy** [**interface** *type number*]
3. **show ipv6 snooping counter** [**interface** *type number*]
4. **show ipv6 snooping features**
5. **show ipv6 snooping policies** [**interface** *type number*]
6. **debug ipv6 snooping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 snooping capture-policy** [**interface** *type number*] **Example:** `Device# show ipv6 snooping capture-policy interface ethernet 0/0`	Displays snooping ND message capture policies.
Step 3	**show ipv6 snooping counter** [**interface** *type number*] **Example:** `Device# show ipv6 snooping counter interface FastEthernet 4/12`	Displays information about the packets counted by the interface counter.
Step 4	**show ipv6 snooping features** **Example:** `Device# show ipv6 snooping features`	Displays information about snooping features configured on the device.
Step 5	**show ipv6 snooping policies** [**interface** *type number*] **Example:** `Device# show ipv6 snooping policies`	Displays information about the configured policies and the interfaces to which they are attached.
Step 6	**debug ipv6 snooping** **Example:** `Device# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

Configuring IPv6 Device Tracking

Configuring IPv6 First-Hop Security Binding Table Content

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor binding** {*ipv6-address* | *ipv6-prefix*} **interface** *type number* [*hardware-address* | *mac-address*][**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*]
4. **ipv6 neighbor binding max-entries** *entries*
5. **ipv6 neighbor binding logging**
6. **exit**
7. **show ipv6 neighbor binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 neighbor binding** {*ipv6-address* | *ipv6-prefix*} **interface** *type number* [*hardware-address* | *mac-address*][**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*] **Example:** `Device(config)# ipv6 neighbor binding 2001:DB8:0:ABCD::1 interface GigabitEthernet 0/0/1 reachable-lifetime 100`	Adds a static entry to the binding table database.
Step 4	**ipv6 neighbor binding max-entries** *entries* **Example:** `Device(config)# ipv6 neighbor binding max-entries 100`	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 5	**ipv6 neighbor binding logging** **Example:** `Device(config)# ipv6 neighbor binding logging`	Enables the logging of binding table main events.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**show ipv6 neighbor binding** **Example:** `Device# show ipv6 neighbor binding`	Displays the contents of a binding table.

Configuring the IPv6 First-Hop Security Binding Table Recovery Mechanism

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor binding** *ipv6-address* **interface** *type number*
4. **ipv6 prefix-list** *list-name* **permit** *ipv6-prefix/prefix-length* **ge** *ge-value*
5. **ipv6 snooping policy** *snooping-policy-id*
6. **destination-glean** {**recovery** | **log-only**} [**dhcp**]
7. **data-glean** {**recovery** | **log-only**} [**ndp** | **dhcp**]
8. **prefix-glean**
9. **protocol dhcp** [**prefix-list** *prefix-list-name*]
10. **exit**
11. **ipv6 destination-guard policy** *policy-name*
12. **enforcement** {**always** | **stressed**}
13. **exit**
14. **interface** *type number*
15. **ipv6 snooping attach-policy** *snooping-policy*
16. **ipv6 destination-guard attach-policy** *policy-name*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 neighbor binding** *ipv6-address* **interface** *type number* **Example:** `Device(config)# ipv6 neighbor binding 2001:db8::1 interface Gigabitethernet3/0/1`	Adds a static entry to the binding table database.
Step 4	**ipv6 prefix-list** *list-name* **permit** *ipv6-prefix/prefix-length* **ge** *ge-value* **Example:** `Device(config)# ipv6 prefix-list abc permit 2001:DB8::/64 ge 128`	Creates an entry in an IPv6 prefix list.
Step 5	**ipv6 snooping policy** *snooping-policy-id* **Example:** `Device(config)# ipv6 snooping policy xyz`	Enters IPv6 snooping configuration mode and allows you to modify the configuration of the snooping policy specified.
Step 6	**destination-glean** {**recovery** | **log-only**} [**dhcp**] **Example:** `Device(config-ipv6-snooping)# destination-glean recovery dhcp`	Specifies that destination addresses should be recovered from DHCP. **Note** If logging (without recovery) is required, use the **destination-glean log-only** command.
Step 7	**data-glean** {**recovery** | **log-only**} [**ndp** | **dhcp**] **Example:** `Device(config-ipv6-snooping)# data-glean recovery ndp`	Enables IPv6 first-hop security binding table recovery using source (or "data") address gleaning. **Note** If logging (without recovery) is required, use the **data-glean log-only** command.
Step 8	**prefix-glean** **Example:** `Device(config-ipv6-snooping)# prefix-glean`	Enables the device to glean prefixes from IPv6 router advertisements (RAs) or Dynamic Host Configuration protocol (DHCP)
Step 9	**protocol dhcp** [**prefix-list** *prefix-list-name*] **Example:** `Device(config-ipv6-snooping)# protocol dhcp prefix-list abc`	(Optional) Specifies that addresses should be gleaned with DHCP and associates the protocol with a specific IPv6 prefix list.
Step 10	**exit** **Example:** `Device(config-ipv6-snooping)# exit`	Exits IPv6 snooping configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**ipv6 destination-guard policy** *policy-name* **Example:** `Device(config)# ipv6 destination-guard policy xyz`	(Optional) Enters destination guard configuration mode and allows you to modify the configuration of the specified destination guard policy.
Step 12	**enforcement** {**always** | **stressed**} **Example:** `Device(config-destguard)# enforcement stressed`	Sets the enforcement level of the policy to be either enforced under all conditions or only when the system is under stress.
Step 13	**exit** **Example:** `Device(config-destguard)# exit`	Exits destination guard configuration mode and returns to global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 0/0/1`	Enters interface configuration mode.
Step 15	**ipv6 snooping attach-policy** *snooping-policy* **Example:** `Device(config-if)# ipv6 snooping attach-policy xyz`	Attaches the IPv6 snooping policy to the interface.
Step 16	**ipv6 destination-guard attach-policy** *policy-name* **Example:** `Device(config-if)# ipv6 destination-guard attach-policy xyz`	Attaches the destination guard policy to the specified interface. **Note** For information about how to configure an IPv6 destination guard policy, see the "IPv6 Destination Guard" module.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Gleaning and Associating Recovery Protocols with Prefix Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 snooping policy** *snooping-policy-id*
4. **protocol** {**dhcp** | **ndp**} [**prefix-list** *prefix-list-name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 snooping policy** *snooping-policy-id* **Example:** `Device(config)# ipv6 snooping policy 200`	Enters IPv6 snooping configuration mode and allows you to modify the configuration of the snooping policy specified.
Step 4	**protocol** {**dhcp** | **ndp**} [**prefix-list** *prefix-list-name*] **Example:** `Device(config-ipv6-snooping)# protocol dhcp prefix-list dhcp_prefix_list`	Specifies that address should be gleaned with dynamic Host Configuration Protocol (DHCP) and associates a recovery protocol (DHCP) with the prefix list.
Step 5	**end** **Example:** `Device(config-ipv6-snooping)# end`	Exits IPv6 snooping configuration mode and returns to privileged EXEC mode.

Configuring IPv6 Device Tracking

Perform this task to provide fine tuning for the life cycle of an entry in the binding table for the IPv6 Device Tracking feature. For IPv6 device tracking to work, the binding table needs to be populated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor tracking** [**retry-interval** *value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 neighbor tracking** [**retry-interval** *value*] **Example:** `Device(config)# ipv6 neighbor tracking`	Tracks entries in the binding table.

Configuring IPv6 Prefix Glean

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 snooping policy** *snooping-policy*
4. **prefix-glean** [**only**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 snooping policy** *snooping-policy* **Example:** `Device(config)# ipv6 snooping policy policy1`	Configures an IPv6 snooping policy and enters IPv6 snooping policy configuration mode.
Step 4	**prefix-glean** [**only**] **Example:** `Device(config-ipv6-snooping)# prefix-glean`	Enables the device to glean prefixes from IPv6 RAs or DHCPv6 traffic.

Configuration Examples for IPv6 Snooping

Example: Configuring IPv6 ND Inspection on an Interface

```
Device(config)# ipv6 snooping policy policy1
Device(config-ipv6-snooping)# exit
Device(config)# interface Gigabitethernet 0/0/1
Device(config-if)# ipv6 snooping attach-policy policy1
.
.
.
Device# show ipv6 snooping policies interface gigabitethernet 0/0/1
Target               Type  Policy              Feature        Target range
Gi0/0/1              PORT  my_policy           Destination Gu vlan all
Gi0/0/1              PORT  policy1             Snooping       vlan all
```

Example: Configuring IPv6 Binding Table Content

```
Device(config)# ipv6 neighbor binding 2001:DB8:0:ABCD::1 interface GigabitEthernet 0/0/1
reachable-lifetime 100
Device(config)# ipv6 neighbor binding max-entries 100
Device(config)# ipv6 neighbor binding logging
Device(config)# exit
```

Example: Configuring IPv6 First-Hop Security Binding Table Recovery

```
Device> enable
Device# configure terminal
Device(config)# ipv6 neighbor binding 2001:db8::1 interface Gigabitethernet3/0/1
Device(config)# ipv6 prefix-list abc permit 2001:DB8::/64 ge 128
Device(config)# ipv6 snooping policy xyz
Device(config-ipv6-snooping)# destination-glean recovery dhcp
Device(config-ipv6-snooping)# data-glean recovery ndp
Device(config-ipv6-snooping)# prefix-glean
Device(config-ipv6-snooping)# protocol dhcp prefix-list abc
Device(config-ipv6-snooping)# exit
Device(config)# ipv6 destination-guard policy xyz
Device(config-destguard)# enforcement stressed
Device(config-destguard)# exit
Device(config)# interface Gigabitethernet 0/0/1
Device(config-if)# ipv6 snooping attach-policy xyz
Device(config-if)# ipv6 destination-guard attach-policy xyz
Device(config-if)# end
```

Example: Configuring Address Gleaning and Associating Recovery Protocols with Prefix Lists

The following example shows that NDP will be used for the recovery for all addresses and that DHCP will be used to recover addresses that match the prefix list called dhcp_prefix_list:

```
Device(config-ipv6-snooping)# protocol ndp
Device(config-ipv6-snooping)# protocol dhcp prefix-list dhcp_prefix_list
```

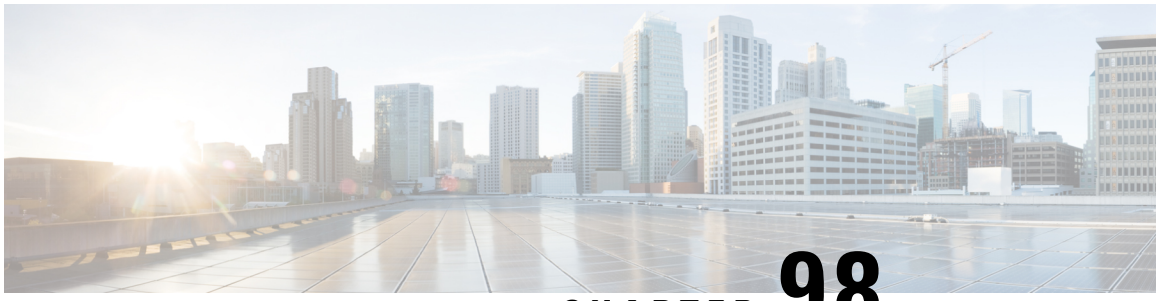
Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 132: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.



CHAPTER 98

IPv6 DAD Proxy

IPv6 Duplicate Address Detection (DAD) Proxy feature responds to the DAD queries on behalf of a node that owns the queried address. It is useful in environments where nodes cannot communicate directly on the link.

Restrictions for IPv6 DAD Proxy

- The IPv6 Duplicate Address Detection (DAD) Proxy feature is not supported on Etherchannel ports.

Information About IPv6 DAD Proxy

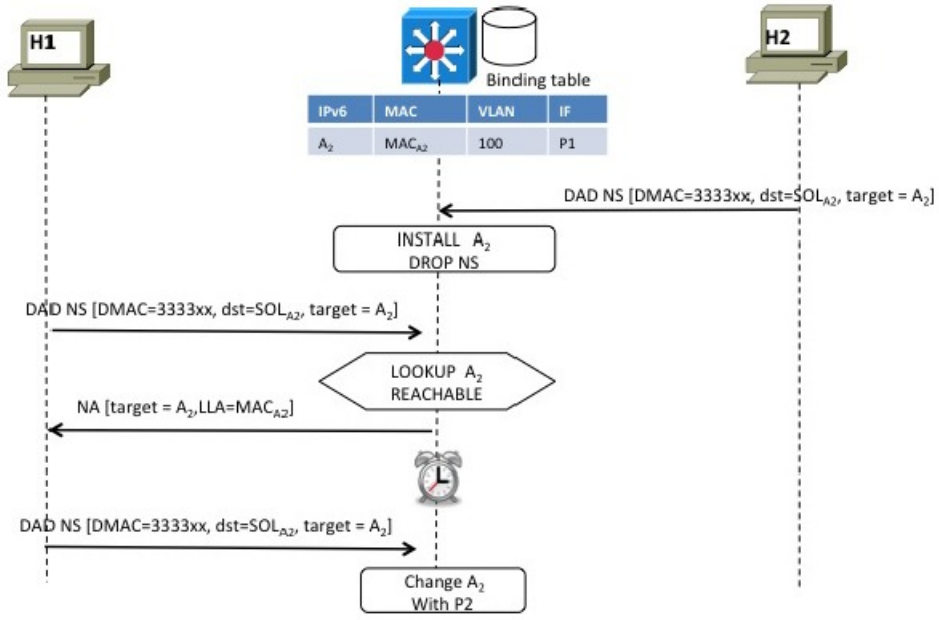
Overview of IPv6 DAD Proxy

The IPv6 Duplicate Address Detection (DAD) feature ensures that all the IP addresses assigned on a particular segment are unique. The process operates when IPv6 hosts directly communicate with one another where hosts cannot communicate directly, and a proxy is required.

After a host verifies that its address is unique, it enables the DAD procedure. However, when two hosts cannot communicate with each other, this procedure cannot detect a duplicate address. If the DAD procedure cannot run, both the hosts assigns the same link-local address, which causes both hosts to fail when they try to reach the Dynamic Host Configuration Protocol version 6 (DHCPv6)server. The IPv6 DAD Proxy feature responds on behalf of the address owner when an address is in use.

The following figure provides an overview of the IPv6 DAD Proxy feature:

Figure 28: IPv6 DAD Proxy



336590

How to Configure IPv6 DAD Proxy

Configuring IPv6 DAD Proxy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **[no] ipv6 nd dad-proxy**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**[no] ipv6 nd dad-proxy** **Example:** `Device(config-if)# ipv6 nd dad-proxy`	Specifies if the ND suppress must operate in DAD-proxy mode. In this mode, the DAD messages are not forwarded. They respond to an existing entry or are added to the binding table.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits router interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 DAD Proxy

Example: Configuring IPv6 DAD Proxy

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet 0/0
Device(config-if)# ipv6 nd dad-proxy
Device(config-if)# end
```

Additional References for IPv6 DAD Proxy

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

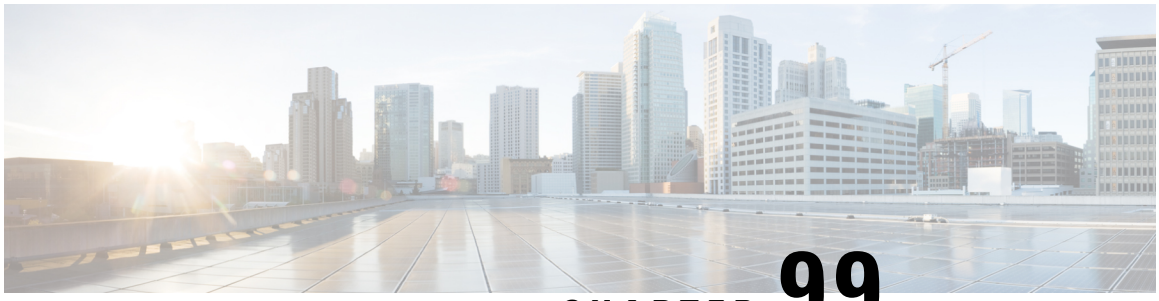
Feature Information for IPv6 DAD Proxy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 133: Feature Information for IPv6 DAD Proxy

Feature Name	Releases	Feature Information
IPv6 DAD Proxy		The following commands were introduced or modified: **ipv6 nd dad-proxy**, **mode dad-proxy**, **mode md-proxy**.



CHAPTER 99

IPv6 Neighbor Discovery Multicast Suppress

IPv6 Neighbor Discovery (ND) Multicast Suppress suppresses the ND multicast Neighbor Solicit (NS) messages, by either dropping it (and responding to solicitations on behalf of the targets) or converting it into unicast traffic. The conversion of multicast traffic into unicast traffic is performed by replacing a Layer-2 Multicast Destination MAC with a Layer- 2 Unicast Destination MAC. This requires the knowledge of addresses on the link and their binding to the Layer-2. The multicast messages suppressed are Neighbor Solicitation (NS) messages.

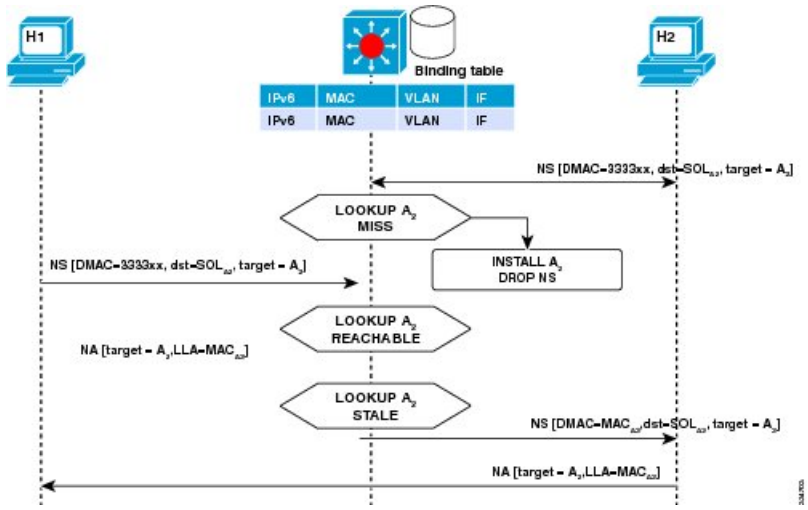
Information About IPv6 Neighbor Discovery Multicast Suppress

Overview of IPv6 Neighbor Discovery Multicast Suppress

The IPv6 Neighbor Discovery (ND) multicast suppress feature stops the ND multicast Neighbor Solicit (NS) messages by dropping them (and responding to solicitations on behalf of the targets) or by converting them into unicast traffic. This feature reduces the amount of control traffic necessary for proper link operations.

When an address is inserted into the binding table, an address resolution request sent to a multicast address is intercepted, and the device either responds on behalf of the address owner or converts the request into a unicast message and forwards it to its destination.

The following figure provides an overview of this feature:



How to Configure IPv6 Neighbor Discovery Multicast Suppress

Configuring IPv6 Neighbor Discovery Multicast Suppress on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd suppress policy** *policy-name*
4. **[no] mode mc-proxy**
5. **[no] mode full-proxy**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd suppress policy** *policy-name* **Example:**	Specifies a name for the Neighbor Discovery (ND) suppress policy to be configured.

	Command or Action	Purpose
	`Device (config)# ipv6 nd suppress policy policy1` `Device (config-nd-suppress)#`	
Step 4	**[no] mode mc-proxy** **Example:** `Device (config-nd-suppress)# mode mc-proxy`	Specifies if the ND suppress must proxy all multicast Neighbor Solicitation (NS) messages.
Step 5	**[no] mode full-proxy** **Example:** `Device (config-nd-suppress)# mode full-proxy`	Specifies if the ND suppress must proxy both unicast and multicast NS messages.
Step 6	**end** **Example:** `Device (config-nd-suppress)# end`	Exits the ND suppress mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 Neighbor Discovery Multicast Suppress

Example: Configuring IPv6 Neighbor Discovery Suppress on an Interface

```
Device> enable
Device(config)# interface Ethernet 0/0
Device(config-if)# ipv6 nd suppress attach-policy policy1
```

Additional References for IPv6 Neighbor Discovery Multicast Suppress

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 134: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 100

DHCP—DHCPv6 Guard

This module describes the Dynamic Host Configuration Protocol version 6 (DHCPv6) Guard feature. This feature blocks DHCP reply and advertisement messages that originate from unauthorized DHCP servers and relay agents that forward DHCP packets from servers to clients. Client messages or messages sent by relay agents from clients to servers are not blocked. The filtering decision is determined by the device role assigned to the receiving switch port, trunk, or VLAN. In addition, to provide a finer level of filter granularity, messages can be filtered based on the address of the sending server or relay agent, or by the prefixes and addresses ranges listed in the reply message. This functionality helps to prevent traffic redirection or denial of service (DoS).

Restrictions for DHCPv6 Guard

- The DHCPv6 guard feature is not supported on Etherchannel ports.

Information About DHCPv6 Guard

DHCPv6 Guard Overview

The DHCPv6 Guard feature blocks reply and advertisement messages that come from unauthorized DHCP servers and relay agents.

Packets are classified into one of the three DHCP type messages. All client messages are always switched regardless of device role. DHCP server messages are only processed further if the device role is set to server. Further processing of server messages includes DHCP server advertisements (for source validation and server preference) and DHCP server replies (for permitted prefixes).

If the device is configured as a DHCP server, all the messages need to be switched, regardless of the device role configuration.

How to Configure DHCPv6 Guard

Configuring DHCP—DHCPv6 Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit host** *address* **any**
5. **exit**
6. **ipv6 prefix-list** *list-name* **permit** *ipv6-prefix* **128**
7. **ipv6 dhcp guard policy** *policy-name*
8. **device-role** {**client** | **server**}
9. **match server access-list** *ipv6-access-list-name*
10. **match reply prefix-list** *ipv6-prefix-list-name*
11. **preference min** *limit*
12. **preference max** *limit*
13. **trusted-port**
14. **exit**
15. **interface** *type number*
16. **switchport**
17. **exit**
18. **exit**
19. **show ipv6 dhcp guard policy** [*policy-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list acl1`	Defines the IPv6 access list and enters IPv6 access list configuration mode.

	Command or Action	Purpose
Step 4	**permit host** *address* **any** **Example:** `Device(config-ipv6-acl)# permit host FE80::A8BB:CCFF:FE01:F700 any`	Sets the conditions in the named IP access list.
Step 5	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and returns to global configuration mode.
Step 6	**ipv6 prefix-list** *list-name* **permit** *ipv6-prefix* **128** **Example:** `Device(config)# ipv6 prefix-list abc permit 2001:0DB8::/64 le 128`	Creates an entry in an IPv6 prefix list.
Step 7	**ipv6 dhcp guard policy** *policy-name* **Example:** `Device(config)# ipv6 dhcp guard policy pol1`	Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode.
Step 8	**device-role** {**client** | **server**} **Example:** `Device(config-dhcp-guard)# device-role server`	Specifies the device role of the device attached to the target (interface or VLAN).
Step 9	**match server access-list** *ipv6-access-list-name* **Example:** `Device(config-dhcp-guard)# match server access-list acl1`	(Optional) Enables verification of the advertised DHCP server and relay address in inspected messages from the configured authorized server access list. If not configured, this check will be bypassed. An empty access list is treated as a permit.
Step 10	**match reply prefix-list** *ipv6-prefix-list-name* **Example:** `Device(config-dhcp-guard)# match reply prefix-list abc`	(Optional) Enables verification of the advertised prefixes in DHCP reply messages from the configured authorized prefix list. If not configured, this check will be bypassed. An empty prefix list is treated as a permit.
Step 11	**preference min** *limit* **Example:** `Device(config-dhcp-guard)# preference min 0`	(Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed.
Step 12	**preference max** *limit* **Example:** `Device(config-dhcp-guard)# preference max 255`	(Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed.

	Command or Action	Purpose
Step 13	**trusted-port** **Example:** `Device(config-dhcp-guard)# trusted-port`	(Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled.
Step 14	**exit** **Example:** `Device(config-dhcp-guard)# exit`	Exits DHCP guard configuration mode and returns to global configuration mode.
Step 15	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Specifies an interface and enters interface configuration mode.
Step 16	**switchport** **Example:** `Device(config-if)# switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 19	**show ipv6 dhcp guard policy** [*policy-name*] **Example:** `Device# show ipv6 dhcp policy guard pol1`	(Optional) Displays the policy configuration as well as all the interfaces where the policy is applied.

Configuration Examples for DHCPv6 Guard

Example: Configuring DHCP—DHCPv6 Guard

The following example displays a sample configuration for DHCPv6 Guard:

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual and configuration information	*Cisco IOS IP Addressing Services Configuration Guide*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for DHCP—DHCPv6 Guard

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 135: Feature Information for DHCP—DHCPv6 Guard

Feature Name	Releases	Feature Information
DHCP—DHCPv6 Guard		The DHCP—DHCPv6 Guard feature blocks DHCP reply and advertisement messages that originate from unauthorized DHCP servers and relay agents that forward DHCP packets from servers to clients. Client messages or messages sent by relay agents from clients to servers are not blocked. The following commands were introduced or modified: **device-role** , **ipv6 dhcp guard attach-policy (DHCPv6 Guard)**, **ipv6 dhcp guard policy**, **match reply prefix-list**, **match server access-list**, **preference (DHCPv6 Guard)**, **show ipv6 dhcp guard policy**, **trusted-port (DHCPv6 Guard)**.

CHAPTER 101

IPv6 Source Guard and Prefix Guard

IPv6 Source Guard and IPv6 Prefix Guard are Layer 2 snooping features that validate the source of IPv6 traffic. IPv6 Source Guard blocks any data traffic from an unknown source. For example, one that is not already populated in the binding table or previously learned through Neighbor Discovery (ND) or Dynamic Host Configuration Protocol (DHCP) gleaning. IPv6 Prefix Guard prevents home-node sourcing traffic outside of the authorized and delegated traffic.

Information About IPv6 Source Guard and Prefix Guard

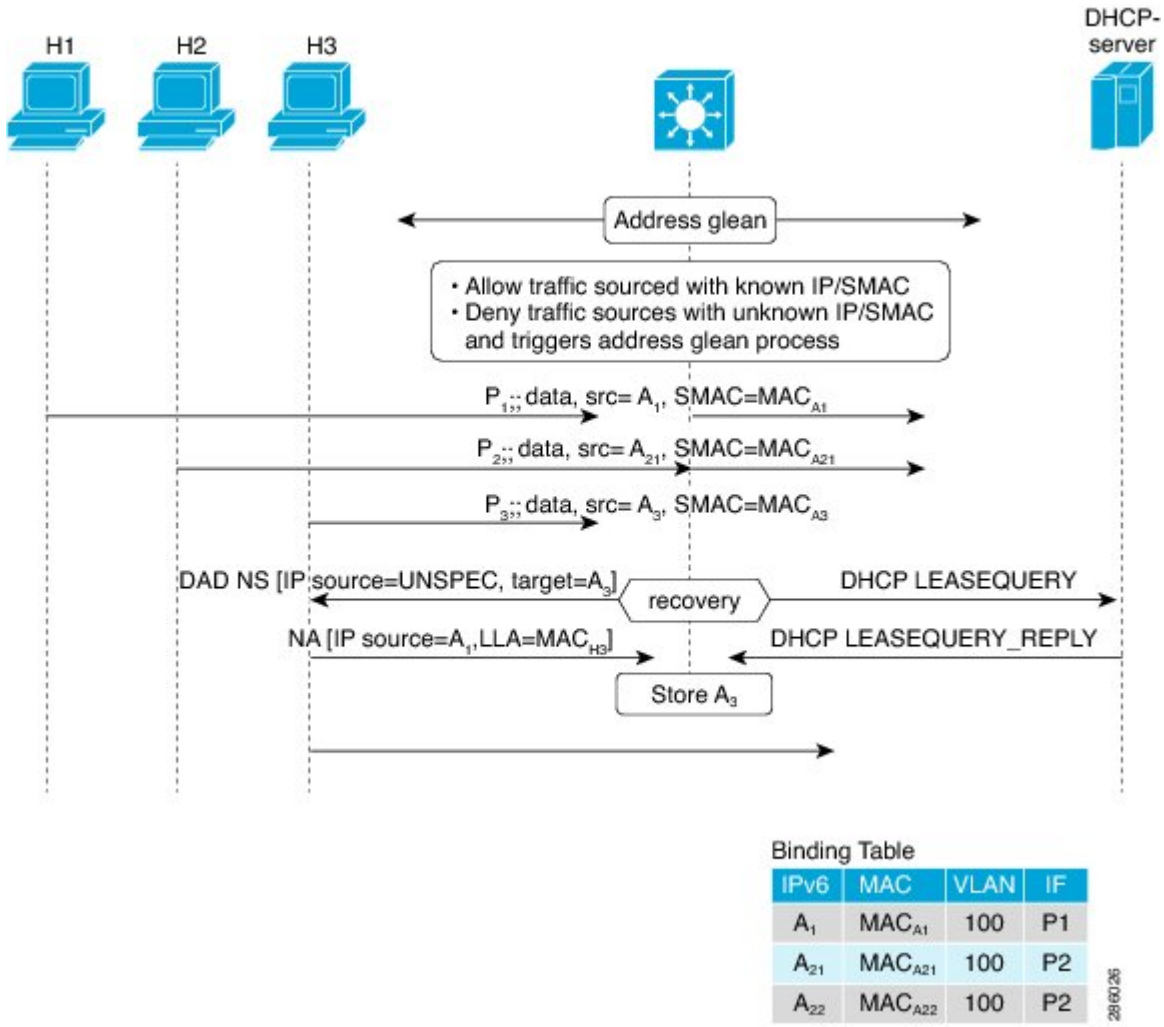
IPv6 Source Guard Overview

IPv6 source guard is an interface feature between the populated binding table and data traffic filtering. This feature enables the device to deny traffic when it is originated from an address that is not stored in the binding table. IPv6 source guard does not inspect ND or DHCP packets; rather, it works in conjunction with IPv6 neighbor discovery (ND) inspection or IPv6 address glean, both of which detect existing addresses on the link and store them into the binding table. IPv6 source guard is an interface between the populated binding table and data traffic filtering, and the binding table must be populated with IPv6 prefixes for IPv6 source guard to work.

IPv6 source guard can deny traffic from unknown sources or unallocated addresses, such as traffic from sources not assigned by a DHCP server. When traffic is denied, the IPv6 address glean feature is notified so that it can try to recover the traffic by querying the DHCP server or by using IPv6 ND. The data-glean function prevents the device and end user from getting deadlocked, whereupon a valid address fails to be stored into the binding table, there is no recovery path, and the end user is unable to connect.

The following illustration provides an overview of how IPv6 source guard works with IPv6 address glean.

Figure 29: IPv6 Source Guard and Address Glean Overview



IPv6 Prefix Guard Overview

The IPv6 Prefix Guard feature works within the IPv6 Source Guard feature, enabling the device to deny traffic originated from nontopologically correct addresses. IPv6 prefix guard is often used when IPv6 prefixes are delegated to devices (for example, home gateways) using DHCP prefix delegation. The feature discovers ranges of addresses assigned to the link and blocks any traffic sourced with an address outside this range.

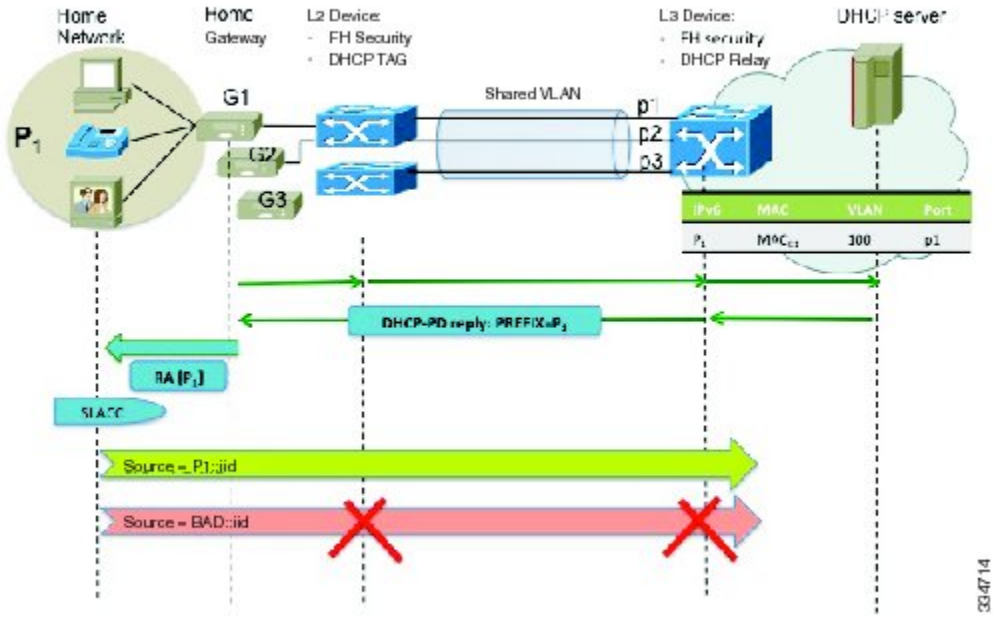
To determine which prefixes should be allowed and which prefixes should be blocked, IPv6 prefix guard uses the following:

- Prefix glean in Router Advertisements (RAs)
- Prefix glean in DHCP prefix delegation
- Static configuration

Whenever a prefix is to be allowed, IPv6 prefix guard downloads it to the hardware table. Whenever a packet is switched, the hardware matches the source of the packet against this table and drops the packet if no match is found.

The following figure shows a service provider (SP) scenario in which prefixes are gleaned in DHCP-PD messages.

Figure 30: Prefixes Gleaned in DHCP-PD Messages Scenario



How to Configure IPv6 Source Guard and Prefix Guard

Configuring IPv6 Source Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 source-guard policy** *source-guard-policy*
4. **permit link-local**
5. **deny global-autoconf**
6. **trusted**

7. **exit**
8. **show ipv6 source-guard policy** [*snooping-policy*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-guard policy** *source-guard-policy* **Example:** `Device(config)# ipv6 source-guard policy my_sourceguard_policy`	Defines an IPv6 source-guard policy name and enters switch integrated security features source-guard policy configuration mode.
Step 4	**permit link-local** **Example:** `Device(config-sisf-sourceguard)# permit link-local`	Allows hardware bridging for all data traffic sourced by a link-local address.
Step 5	**deny global-autoconf** **Example:** `Device(config-sisf-sourceguard)# deny global-autoconf`	Denies data traffic from auto-configured global addresses.
Step 6	**trusted** **Example:** `Device(config-sisf-sourceguard)# trusted`	Allows hardware bridging for all data traffic on the target where the policy is applied.
Step 7	**exit** **Example:** `Device(config-sisf-sourceguard)# exit`	Exits source-guard policy configuration mode and returns to privileged EXEC mode.
Step 8	**show ipv6 source-guard policy** [*snooping-policy*] **Example:** `Device# show ipv6 source-guard policy policy1`	Displays the IPv6 source-guard policy configuration.

Configuring IPv6 Source Guard on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 source-guard attach-policy** *source-guard-policy*
5. **exit**
6. **show ipv6 source-guard policy** *source-guard-policy*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 3/13`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 source-guard attach-policy** *source-guard-policy* **Example:** `Device(config-if)# ipv6 source-guard attach-policy my_source_guard_policy`	Applies IPv6 source guard on an interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and places the device in privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *source-guard-policy* **Example:** `Device# show ipv6 source-guard policy policy1`	Displays all the interfaces on which IPv6 source guard is applied.

Configuring IPv6 Prefix Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 source-guard policy** *source-guard-policy*
4. **validate address**
5. **validate prefix**
6. **exit**
7. **show ipv6 source-guard policy** [*source-guard-policy*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-guard policy** *source-guard-policy* **Example:** `Device(config)# ipv6 source-guard policy my_snooping_policy`	Defines an IPv6 source-guard policy name and enters switch integrated security features source-guard policy configuration mode.
Step 4	**validate address** **Example:** `Device(config-sisf-sourceguard)# no validate address`	Disables the validate address feature and enables the IPv6 prefix guard feature to be configured.
Step 5	**validate prefix** **Example:** `Device(config-sisf-sourceguard)# validate prefix`	Enables IPv6 source guard to perform the IPv6 prefix-guard operation.
Step 6	**exit** **Example:** `Device(config-sisf-sourceguard)# exit`	Exits switch integrated security features source-guard policy configuration mode and returns to privileged EXEC mode.
Step 7	**show ipv6 source-guard policy** [*source-guard-policy*] **Example:** `Device# show ipv6 source-guard policy policy1`	Displays the IPv6 source-guard policy configuration.

Configuration Examples for IPv6 Source Guard and Prefix Guard

Example: Configuring IPv6 Source Guard and Prefix Guard

```
Device# ipv6 source-guard policy policy1

Policy guard configuration:
  validate prefix
  validate address
```

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 136: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 102

IPv6 Destination Guard

The IPv6 Destination Guard feature works with IPv6 neighbor discovery to ensure that the device performs address resolution only for those addresses that are known to be active on the link. It relies on the address glean functionality to populate all destinations active on the link into the binding table and then blocks resolutions before they happen when the destination is not found in the binding table.

Prerequisites for IPv6 Destination Guard

- You should be familiar with the IPv6 Neighbor Discovery feature. For information about IPv6 neighbor discovery, see the “Implementing IPv6 Addressing and Basic Connectivity” module.
- You should be familiar with the IPv6 First-Hop Security Binding Table feature. For information, see the “IPv6 First-Hop Security Binding Table” module.

Information About IPv6 Destination Guard

IPv6 Destination Guard Overview

The IPv6 Destination Guard feature works with IPv6 neighbor discovery to ensure that the device performs address resolution only for those addresses that are known to be active on the link. It relies on the address glean functionality to populate all destinations active on the link into the binding table and then blocks resolutions before they happen when the destination is not found in the binding table.

Prior to filtering incoming routed traffic, the device gleans addresses on the link, by snooping Neighbor Discovery Protocol (NDP) and DHCP messages. When a packet reaches the device and there is not yet an adjacency for the destination or for the next hop, the NDP consults the device binding table to verify that the destination on link or the next-hop have been previously gleaned. If the destination is not found in the binding table, the packet is dropped. Otherwise, neighbor discovery resolution is performed.

How to Configure the IPv6 Destination Guard

Configuring IPv6 Destination Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 destination-guard policy** *policy-name*
4. **enforcement** {**always** | **stressed**}
5. **exit**
6. **interface** *type number*
7. **ipv6 destination-guard attach-policy** [*policy-name*]
8. **exit**
9. **show ipv6 destination-guard policy** [*policy-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 destination-guard policy** *policy-name* **Example:** `Device(config)# ipv6 destination-guard policy pol1`	Defines the destination guard policy name and enters destination-guard configuration mode.
Step 4	**enforcement** {**always**	**stressed**} **Example:** `Device(config-destguard)# enforcement always`	Sets the enforcement level for the target address.
Step 5	**exit** **Example:** `Device(config-destguard)# exit`	Exits destination-guard configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.
Step 7	**ipv6 destination-guard attach-policy** [*policy-name*] **Example:** `Device(config-if)# ipv6 destination-guard attach-policy pol1`	Attaches a destination guard policy to an interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC configuration mode.
Step 9	**show ipv6 destination-guard policy** [*policy-name*] **Example:** `Device# show ipv6 destination-guard policy pol1`	(Optional) Displays the policy configuration and all interfaces where the policy is applied.

Configuration Examples for IPv6 Destination Guard

Example: Configuring an IPv6 Destination Guard Policy

The following example shows how to configure a destination guard policy:

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet 0/0/1
Router(config-if)# ipv6 destination-guard attach-policy destination

Router# show ipv6 destination-guard policy destination
Destination guard policy Destination:
  enforcement always
        Target: Gi0/0/1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 137: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 103

IPv6 RFCs

Standards and RFCs

RFCs	Title
RFC 1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
RFC 1267	*A Border Gateway Protocol 3 (BGP-3)*
RFC 1305	*Network Time Protocol (Version 3) Specification, Implementation and Analysis*
RFC 1583	*OSPF version 2*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1886	*DNS Extensions to Support IP version 6*
RFC 1918	*Address Allocation for Private Internets*
RFC 1981	*Path MTU Discovery for IP version 6*
RFC 2080	*RIPng for IPv6*
RFC 2281	*Cisco Hot Standby Router Protocol (HSRP)*
RFC 2332	*NBMA Next Hop Resolution Protocol (NHRP)*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2375	*IPv6 Multicast Address Assignments*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2402	*IP Authentication Header*
RFC 2404	*The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header*
RFC 2406	*IP Encapsulating Security Payload (ESP)*

RFCs	Title
RFC 2407	*The Internet Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*Internet Key Exchange (IKE)*
RFC 2427	*Multiprotocol Interconnect over Frame Relay*
RFC 2428	*FTP Extensions for IPv6 and NATs*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2461	*Neighbor Discovery for IP Version 6 (IPv6)*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2463	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 2464	*Transmission of IPv6 Packets over Ethernet*
RFC 2467	*Transmission of IPv6 Packets over FDDI*
RFC 2472	*IP Version 6 over PPP*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2474	*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2475	*An Architecture for Differentiated Services Framework*
RFC 2492	*IPv6 over ATM*
RFC 2545	*Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Specification*
RFC 2597	*Assured Forwarding PHB*
RFC 2598	*An Expedited Forwarding PHB*
RFC 2640	*Internet Protocol, Version 6 Specification*
RFC 2684	*Multiprotocol Encapsulation over ATM Adaptation Layer 5*
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*
RFC 2710	*Multicast Listener Discovery (MLD) for IPv6*
RFC 2711	*IPv6 Router Alert Option*
RFC 2732	*Format for Literal IPv6 Addresses in URLs*

RFCs	Title
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation-Protocol Translation (NAT-PT)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 3068	*An Anycast Prefix for 6to4 Relay Routers*
RFC 3095	*RObust Header Compression (ROHC): Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed*
RFC 3107	*Carrying Label Information in BGP-4*
RFC 3137	*OSPF Stub Router Advertisement*
RFC 3147	*Generic Routing Encapsulation over CLNS*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*
RFC 3315	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3319	*Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3484	*Default Address Selection for Internet Protocol version 6 (IPv6)*
RFC 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3576	*Change of Authorization*
RFC 3587	*IPv6 Global Unicast Address Format*
RFC 3590	*Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
RFC 3596	*DNS Extensions to Support IP Version 6*
RFC 3633	*DHCP IPv6 Prefix Delegation*
RFC 3646	*DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3697	*IPv6 Flow Label Specification*
RFC 3736	*Stateless DHCP Service for IPv6*

RFCs	Title
RFC 3756	*IPv6 Neighbor Discovery (ND) Trust Models and Threats*
RFC 3759	*RObust Header Compression (ROHC): Terminology and Channel Mapping Examples*
RFC 3775	*Mobility Support in IPv6*
RFC 3810	*Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
RFC 3846	*Mobile IPv4 Extension for Carrying Network Access Identifiers*
RFC 3879	*Deprecating Site Local Addresses*
RFC 3898	*Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*
RFC 3956	*Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
RFC 3963	*Network Mobility (NEMO) Basic Support Protocol*
RFC 3971	*SEcure Neighbor Discovery (SEND)*
RFC 3972	*Cryptographically Generated Addresses (CGA)*
RFC 4007	*IPv6 Scoped Address Architecture*
RFC 4075	*Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*
RFC 4087	*IP Tunnel MIB*
RFC 4091	*The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework*
RFC 4092	*Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)*
RFC 4109	*Algorithms for Internet Key Exchange version 1 (IKEv1)*
RFC 4191	*Default Router Preferences and More-Specific Routes*
RFC 4193	*Unique Local IPv6 Unicast Addresses*
RFC 4214	*Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*
RFC 4242	*Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 4282	*The Network Access Identifier*
RFC 4283	*Mobile Node Identifier Option for Mobile IPv6*
RFC 4285	*Authentication Protocol for Mobile IPv6*
RFC 4291	*IP Version 6 Addressing Architecture*

RFCs	Title
RFC 4292	*IP Forwarding Table MIB*
RFC 4293	*Management Information Base for the Internet Protocol (IP)*
RFC 4302	*IP Authentication Header*
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4308	*Cryptographic Suites for IPsec*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*
RFC 4382	*MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base*
RFC 4443	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 4552	*Authentication/Confidentiality for OSPFv3*
RFC 4594	*Configuration Guidelines for DiffServ Service Classes*
RFC 4601	*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification*
RFC 4610	*Anycast-RP Using Protocol Independent Multicast (PIM)*
RFC 4649	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option*
RFC 4659	*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
RFC 4724	*Graceful Restart Mechanism for BGP*
RFC 4798	*Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*
RFC 4818	*RADIUS Delegated-IPv6-Prefix Attribute*
RFC 4861	*Neighbor Discovery for IP version 6 (IPv6)*
RFC 4862	*IPv6 Stateless Address Autoconfiguration*
RFC 4884	*Extended ICMP to Support Multi-Part Messages*
RFC 4885	*Network Mobility Support Terminology*
RFC 4887	*Network Mobility Home Network Models*
RFC 5015	*Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
RFC 5059	*Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*
RFC 5072	*IPv6 over PPP*
RFC 5095	*Deprecation of Type 0 Routing Headers in IPv6*
RFC 5120	*M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*

RFCs	Title
RFC 5130	*A Policy Control Mechanism in IS-IS Using Administrative Tags*
RFC 5187	*OSPFv3 Graceful Restart*
RFC 5213	*Proxy Mobile IPv6*
RFC 5308	*Routing IPv6 with IS-IS*
RFC 5340	*OSPF for IPv6*
RFC 5460	*DHCPv6 Bulk Leasequery*
RFC 5643	*Management Information Base for OSPFv3*
RFC 5838	*Support of Address Families in OSPFv3*
RFC 5844	*IPv4 Support for Proxy Mobile IPv6*
RFC 5845	*Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6*
RFC 5846	*Binding Revocation for IPv6 Mobility*
RFC 5881	*Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*
RFC 5905	*Network Time Protocol Version 4: Protocol and Algorithms Specification*
RFC 5969	*IPv6 Rapid Deployment on IPv4 Infrastructures (6RD) -- Protocol Specification*
RFC 6105	*IPv6 Router Advertisement Guard*
RFC 6620	*FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*

PART X

MACsec and MKA

CHAPTER 104

WAN MACSEC and MKA Support Enhancements

The WAN MACsec and MKA feature introduces MACsec support on WAN, and uplink support and Pre-shared key support for the Macsec Key Agreement protocol (MKA).

Feature Information for WAN MACsec and MKA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 138: Feature Information for WAN MACsec and MKA

Feature Name	Releases	Feature Information
WAN MACsec and MKA	Cisco IOS XE Release 3.14S	The WAN MACsec and MKA feature introduces MACsec support on WAN and uplink support and pre-shared key support for the MACsec Key Agreement protocol (MKA). The following commands were introduced or modified: confidentiality-offset, eapol destination-mac, key-server, linksec policy, replay-protection window-size .
MACsec on WAN Interface Cards	Cisco IOS XE Release 3.16S	The MACsec on WAN Interface Cards feature introduces MACsec support on WAN interface cards on Cisco 4000 Series Integrated Services Routers (ISRs).

Feature Name	Releases	Feature Information
MACsec CLI Option to Change EAPoL Frame Ethernet Type	Cisco IOS XE Release 3.17S	The MACsec CLI Option to Change EAPOL Frame Ethernet Type feature provides a configuration option to allow users to change the Extensible Authentication Protocol over LAN (EAPoL) Frame Ethernet Type. The following commands were introduced or modified: eapol eth-type.
Support for configuring port-channel with MACsec encryption	Cisco IOS XE Gibraltar 17.2	This enhancement lets you configure port-channels on MACsec-enabled interfaces for seamless flow of port-channel traffic; therefore, the traffic is secured.

Prerequisites for WAN MACsec and MKA Support Enhancements

- WAN MACsec requires MACsec license. See Table 8 in the document titled *Cisco ASR 1000 Series Ethernet Line Cards Data Sheet* – https://www.cisco.com/c/en/us/products/collateral/application-networking-services/wide-area-application-services-waas-software/data-sheet-c78-729778.html
- The Cisco ISR 4000 platforms require HSECK9 license to configure MACsec.
- Layer 2 transparent Ethernet Services must be present.
- The service provider network must provide a MACsec Layer 2 Control Protocol transparency such as, Extensible Authentication Protocol over LAN (EAPoL).

Restrictions for WAN MACsec and MKA Support Enhancements

- On Cisco ASR 1000 Series Aggregation Services Routers, MACsec does not support AAA accounting.
- On Cisco ASR 1000 Series Aggregation Services Routers, configuring MKA is not supported in a high availability cluster.
- MACsec is supported up to line rate on each interface. However, the forwarding capability may be limited by the maximum system forwarding capability.
- On the Cisco ASR1001-X router, MACsec is supported on the built-in ports only. It cannot be enabled on a Shared Port Adapter (SPA) that is installed on the router.
- To configure port-channel, ensure that you configure MACsec at each interface of the link bundle.
- MACsec configured on the native subinterface with the command **macsec dot1q-in-clear 1** on the main interface is not supported.
- From Cisco IOS XE Denali 16.3.3 release onwards, during RP Switchover, re-entry of macsec commands in physical/sub-interface configuration mode is not required.

- If the MKA session is torn down because of key unwrap failure, re-configure the pre-shared key based MKA session using MACsec configuration commands on the respective interfaces to bring the MKA session up.
- MACsec-configured on physical interface with Ethernet Virtual Circuits (EVC) is not supported. The EAPoL frames will get dropped in such cases.
- On Cisco ASR 1000 Series Aggregation Services Routers, the following table lists the GigabitEthernet interface and the maximum number of peers that are supported per interface:

GigabitEthernet Interface	Peers per Interface
1G	8
10G	32
40G	60
100G	120

- When `macsec dot1q-in-clear` is enabled, the native VLAN is not supported.

Information About WAN MACsec and MKA Support Enhancements

MACsec and MKA Overview

MACsec is an IEEE 802.1AE standards based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. Only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

The 802.1AE encryption with MACsec Key Agreement (MKA) is supported on downlink ports for encryption between the routers or switches and host devices.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet.

To provide MACsec services over the WAN or Metro Ethernet, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

The packet body in an EAP-over-LAN (EAPOL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participants after 3 hearbeats (each hearbeat is of 2 seconds), peers are deleted from the live peer list For example, if a client disconnects, the participant on the switch continues to operate MKA until 3 heartbeats have elapsed after the last MKPDU is received from the client.

The MKA feature support provides tunneling information such as VLAN tag (802.1Q tag) in the clear so that the service provider can provide service multiplexing such that multiple point to point or multipoint services can co-exist on a single physical interface and differentiated based on the now visible VLAN ID.

In addition to service multiplexing, VLAN tag in the clear also enables service providers to provide quality of service (QoS) to the encrypted Ethernet packet across the SP network based on the 802.1P (CoS) field that is now visible as part of the 802.1Q tag.

Benefits of WAN MACsec and MKA Support Enhancements

- Support for Point-to-point (P2P) deployment models.
- Support for Point-to-Multipoint (P2MP) deployment models.
- Support for multiple P2P and P2MP deployments on the same physical interface.
- Support for 128- and 256-bit Advanced Encryption Standard–Galois Counter Mode (AES-GCM) encryption for data packets.
- Support for 128- and 256-bit Advanced Encryption Standard-Cipher-based Message Authentication Code (AEC-CMAC) encryption for control packets.
- Support for VLAN tag in the clear option to enable Carrier Ethernet Service Multiplexing.
- Support for coexisting of MACsec and Non-MACsec subinterfaces.
- Support for configurable Extensible Authentication Protocol over LAN (EAPoL) destination address.
- Support for configurable option to change the EAPoL Ethernet type.
- Support for configurable replay protection window size to accommodate packet reordering in the service provider network.

Best Practices for Implementing WAN MACsec and MKA Support Enhancements

- Ensure basic Layer 2 Ethernet connectivity is established and verified before attempting to enable MACsec. Basic ping between the customer edge devices must work.
- When you are configuring WAN MACsec for the first time, ensure that you have out of band connectivity to the remote site to avoid locking yourself out after enabling MACsec, if the session fails to establish.
- We recommend that you configure the **access-control should-secure** command while enabling MACsec for the first time and subsequently remove the command to change to default **access-control must-secure**, once the session establishment is successful, unless it is needed for migration.
- We recommend that you configure an interface MTU, adjusting it for MACsec overhead, for example, 32 bytes. Although MACsec encryption and decryption occurs at the physical level and MTU is size does not effect the source or destination router, it may effect the intermediate service provider router. Configuring an MTU value at the interface allows for MTU negotiation that includes MACsec overhead.

MKA Policy Inheritance

On WAN routers, MKA policy is inherited and also it has a default value. When a new session is started, the following rules apply:

- If an MKA policy is configured on a subinterface, it will be applied when an MKA session is started.
- If an MKA policy is not configured on a subinterface, a policy that is configured on the physical interface is be applied at session start.
- If a MKA policy is not configured on a subinterface or physical interface, default policy is applied at session start.

Key Lifetime and Hitless Key Rollover

A MACsec key chain can have multiple pre-shared keys (PSK) each configured with a key id and an optional lifetime. A key lifetime specifies at which time the key expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the key chain after the lifetime is expired. Time zone of the key can be local or UTC. Default time zone is UTC.

Use the **key chain** *name* **macsec** to configure the MACsec key chain.

You can Key rolls over to the next key within the same key chain by configuring a second key in the key chain and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless, that is, key rolls over without traffic interruption.

Note The lifetime of the keys need to be overlapped in order to achieve hitless key rollover.

Encryption Algorithms for Protocol Packets

Cryptographic Algorithm selection for MKA control protocol packets encryption is as follows:

- Cryptographic Algorithm to encrypt MKA control protocol packets is configured as part of the key chain. There can be only one cryptographic algorithm configured per key chain.
- A key server uses the configured MKA cryptographic algorithm from the key chain that is used.
- All nonkey servers must use the same cryptographic algorithm as the key server.

If an MKA cryptographic algorithm is not configured, a default cryptographic algorithm of AES-CMAC-128 (Cipher-based Message Authentication Code with 128-bit Advanced Encryption Standard) is used.

Encryption algorithm for Data packets:

```
mka policy p1
macsec-cipher-suite [gcm-aes-128 | gcm-aes-256
```

Encryption algorithm for MKA Control packets

```
key chain <name> macsec
key 01
```

```
key-string <Hex string>
cryptographic-algorithm [aes-256-cmac  | aes-128-cmac]
```

It is recommended to change data packets cipher suite in the key server for the cipher suite rollover to be seamless, if the nonkey servers have the same cipher-suite configured in the list or is with default configuration.

Access Control Option for Smoother Migration

When MACsec is enabled on an interface, the entire interface traffic is secured, by default. MACsec does not allow any unencrypted packets to be transmitted or received from the same physical interface. However, to enable MACsec on selected subinterfaces, an additional Cisco proprietary extension has been implemented to allow unencrypted packets to be transmitted or received from the same physical interface.

Use the **macsec access-control** {**must-secure** | **should-secure**} command to control the behavior of unencrypted packets.

- The **should-secure** keyword allows unencrypted packets from the physical interface or subinterfaces to be transmitted or received.
- The **must-secure** keyword does not allow unencrypted packets from physical interface or subinterfaces to be transmitted or received. All such packets are dropped except for MKA control protocol packets
- If MACsec is enabled only on selected subinterfaces, configure the **should-secure** keyword option on the corresponding interface.

The default configuration for MACsec on subinterfaces is **macsec access-control must-secure**. This option is enabled by default when the **macsec** command is configured on an interface.

Note The **macsec access-control should-secure** command can be configured only at the interface level and not the subinterface. Configuring this command allows unencrypted traffic on a secured MACsec session.

Note For non-MACsec subinterface, you must configure the **should-secure** option for traffic to pass.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MKA (MACsec Key Agreement) is used as the control protocol. MKA selects the cipher suite to be used for encryption and to exchange the required keys and parameters between peers.

MKA uses Extensible Authentication Protocol over LAN (EAPoL) as the transport protocol to transmit MKA messages. By default, EAPoL uses a destination multicast MAC address of 01:80:c2:00:00:03 to multicast packets to multiple destinations. EAPoL is a standards-based protocol and other authentication mechanisms such as IEEE 802.1X also use the same protocol. Devices in the service provider cloud might consume this packet (based on the destination multicast MAC address), and try to process the EAPoL packet and eventually drop the packet. This causes MKA session to fail.

Use the **eapol destination-address** command to change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider. This ensures that the service provider tunnels the packet like any other data packet instead of consuming them.

Note The EAPoL destination address can be configured independently on either physical or subinterface level. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on the subinterface overrides the inherited value or policy for that subinterface.

Replay Protection Window Size

Replay protection is a feature provided by MACsec to counter replay attacks. Each encrypted packet is assigned a unique sequence number and the sequence is verified at the remote end. Frames transmitted through a Metro Ethernet service provider network are highly susceptible to reordering due to prioritization and load balancing mechanisms used within the network.

A replay window is necessary to support use of MACsec over provider networks that reorder frames. Frames within the window can be received out of order, but are not replay protected. The default window size is set to 64. Use the **macsec replay-protection window-size** command to change the replay window size. The range for window size is 0 to 4294967295.

The replay protection window may be set to zero to enforce strict reception ordering and replay protection.

Note A replay protection window can be configured independently on either physical interface or subinterface. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on subinterface overrides the inherited value or policy for that sub-interface.

MACsec on WAN Interface Cards

In Cisco IOS XE Release 3.16S, MACsec is introduced on WAN interface cards (NIM-2GE-CU-SFP and NIM-2GE-CU-SFP) on Cisco 4000 Series Integrated Services Routers (ISRs).

This WAN interface card is a two one-Gigabit Ethernet-port Next Generation WAN Interface Card.

The following platforms support the Next Generation WAN Interface Card:

- Cisco ISR 4451
- Cisco ISR4431
- Cisco ISR4351
- Cisco ISR 4331
- Cisco ISR 4321

OIR Support

When a WAN interface card is operationally inserted or removed (OIR), the configuration associated with that interface is preserved such that if the interface is ever reinserted into the system it appears with the same configuration. However, in Cisco IOS XE Release 3.16s on Cisco ISR routers the following limitations apply for MACsec and MKA sessions:

- In some scale scenarios, after OIR MKA/MACsec session may be lost.

- MKA/MACsec session must be reestablished after OIR.

MACsec Performance on Cisco 4000 Series Integrated Services Routers

Table 139: Performance Numbers on Cisco ISR 4451 Router

Frame Size	NDR per Port (pps)	Line Rate (%)	Module CPU (%)	Host CPU (%)
64	1,077,532	72.41	44	65
128	692,568	82	29	42
256	405,797	89.6	17	25
iMIX	296,500	90.57	13	24
512	221,615	94.32	9	14
1024	116,163	97.02	5	7
1518	79,609	97.95	3.5	5
9000	13,808	99.64%	1	2

MACsec Performance on Cisco ASR 1000 Platforms

The following tables show the performance numbers on Cisco ASR 1000 routers from Cisco IOS XE 16.6 release onwards.

Table 140: Performance Numbers on Cisco ASR1001-X Router

Frame Size	Aggregate Rate Bits (bps)	Line Rate per port (%)	ESP CPU (%)
64	10064767891.17	65.59	93.33
iMIX	17763891467.40	93.14	26
1418	19311044388.60	97.89	9

Table 141: Performance Numbers on Cisco ASR1001-HX Router

Frame Size	Aggregate Rate Bits (bps)	Line Rate per port (%)	ESP CPU (%)
64	28681245486.53	65.59	99
iMIX	65019905182.40	93.14	42
1418	64975057119.60	97.89	11

Table 142: Performance Numbers on Cisco ASR1002-HX Router

Frame Size	Aggregate Rate Bits (bps)	Line Rate per port (%)	ESP CPU (%)
64	51467063849.50	65.59	96
iMIX	105267526427	93.14	36
1418	100007152449	97.89	10

MACsec Compatibility Matrix for ASR 1000 and ISR 4400 Platforms

Platform	Built-In Ports	EPA-18x1GE	EPA-10x10GE	EPA-1x40GE / EPA-2x40GE	NIM-2GE-CU-SFP
ASR1001-X	Cisco IOS XE Release 3.13.1S	NA	NA	NA	NA
ASR1001-HX	Cisco IOS XE Everest Release 16.4.1	NA	NA	NA	NA
ASR1002-HX	Cisco IOS XE Denali Release 16.3.1	Cisco IOS XE Denali Release 16.3.1	Cisco IOS XE Denali Release 16.3.2 / 16.4.1	Cisco IOS XE Fuji Release 16.8.1	NA
ASR1006-X	NA	Cisco IOS XE Everest Release 16.4.1	Cisco IOS XE Denali Release 16.3.1	Cisco IOS XE Fuji Release 16.8.1	NA
ASR1009-X	NA	Cisco IOS XE Everest Release 16.4.1	Cisco IOS XE Denali Release 16.3.1	Cisco IOS XE Fuji Release 16.8.1	NA
ASR1013	NA	Cisco IOS XE Everest Release 16.4.1	Cisco IOS XE Denali Release 16.3.1	Cisco IOS XE Fuji Release 16.8.1	NA
ISR44XX	NA	NA	NA	NA	Cisco IOS XE Release 3.16.0S
ISR43XX	NA	NA	NA	NA	Cisco IOS XE Release 3.16.0S
ISR4462	Cisco IOS XE Fuji Release 16.9.1	NA	NA	NA	Cisco IOS XE Release 3.16.0S

Note

- GLC-100FX is not supported.
- MIP-100 is required for ASR1006X, ASR1009X, and ASR1013 platforms for EPA18x1GE, EPA-10x10GE, EPA-1x40GE, and EPA-2x40GE.
- MACsec on ASR1001-X requires IPsec license.
- MACsec on ASR1001-HX, ASR1002-HX, and EPAs require per port MACsec licenses.
- The Cisco ISR 4000 platforms require HSECK9 license to configure MACsec.

Note Starting from IOS XE 17.2 Gibraltar, port-channel configuration is supported with MACsec. To configure this feature, ensure that you configure MACsec at each interface of the link bundle. For more information, see *Configuration Examples*

How to Configure WAN MACsec and MKA Support Enhancements

Configuring MKA

The MACsec Key Agreement (MKA) enables configuration and control of keying parameters. Perform the following task to configure MKA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **include-icv-indicator**
5. **key-server priority** *key-server-priority*
6. **macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256** | **gcm-aes-xpn-128** | **gcm-aes-xpn-256**}
7. **sak-rekey interval** *interval*
8. **confidentiality-offset 30**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mka policy** *policy-name* **Example:** `Device(config)# mka policy MKAPolicy`	Configures an MKA policy.
Step 4	**include-icv-indicator** **Example:** `Device(config-mka-policy)# include-icv-indicator`	(Optional) Include ICV indicator in MKPDU.
Step 5	**key-server priority** *key-server-priority* **Example:** `Device(config-mka-policy)# key-server priority 200`	(Optional) Configures MKA key server priority.
Step 6	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256** | **gcm-aes-xpn-128** | **gcm-aes-xpn-256**} **Example:** `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128 gcm-aes-256`	(Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order.
Step 7	**sak-rekey interval** *interval* **Example:** `Device(config-mka-policy)# sak-rekey interval 30`	(Optional) Sets the SAK rekey interval (in seconds). The range is from 30 to 65535, and the default value is 0. The SAK rekey timer does not start by default until it is configured. • To stop the SAK rekey timer, use the **no sak-rekey interval** command under the defined MKA policy.
Step 8	**confidentiality-offset 30** **Example:** `Device(config-mka-policy)# confidentiality-offset 30`	(Optional) Configures confidentiality offset for MACsec operation.
Step 9	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-mka-policy)# end`	**Note** The MKA policy does not process confidentiality offset for XPN ciphers. Therefore when both XPN and non-XPN ciphers are configured in an MKA policy alongwith confidentiality offset, the confidentiality offset is ignored for XPN ciphers. It is therefore strongly recommended to use your discretion while using configuring a MKA policy with XPN or non-XPN ciphers.

Example

You can use the **show mka policy** command to verify the configuration. Here's a sample output of the **show** command. If you do not want to include icv-indicator in MKPDUs, use the **no include-icv-indicator** command in the MKA policy.

```
MKA Policy Summary...

Codes : CO - Confidentiality Offset, ICVIND - Include ICV-Indicator,
        SAKR OLPL - SAK-Rekey On-Live-Peer-Loss,
        DP - Delay Protect, KS Prio - Key Server Priority

Policy            KS   DP    CO SAKR  ICVIND Cipher          Interfaces
Name              Prio          OLPL         Suite(s)        Applied
================================================================================
*DEFAULT POLICY*  0    FALSE 0  FALSE TRUE   GCM-AES-128     N/A
                                              GCM-AES-256

confid50          0    FALSE 50 FALSE TRUE   GCM-AES-128
                                              GCM-AES-256

icv               0    FALSE 0  FALSE TRUE   GCM-AES-128     Te3/0/9
                                              GCM-AES-256

k10               0    FALSE 0  FALSE TRUE   GCM-AES-128
                                              GCM-AES-256

xpn128            0    FALSE 0  FALSE TRUE   GCM-AES-XPN-128 Fo2/1/1
```

Configuring MACsec and MKA on Interfaces

Perform the following task configure MACsec and MKA on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mka policy** *policy-name*
5. **mka pre-shared-keykey-chain***key-chain-name*
6. **macsec**

7. **macsec replay-protection window-size**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Enters interface configuration mode.
Step 4	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy
Step 5	**mka pre-shared-keykey-chain***key-chain-name* **Example:** `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	Configures an MKA pre-shared-key key-chain keychain1 **Note** The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.
Step 6	**macsec** **Example:** `Device(config-if)# macsec`	Configures MACsec for the EAPOL frame ethernet type.
Step 7	**macsec replay-protection window-size** **Example:** `Device(config-if)# macsec replay-protection window-size 10`	Sets the MACsec window size for replay protection.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring MKA Pre-shared Key

Perform the following task to configure MACsec Key Agreement (MKA) pre-shared key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *key-chain-name* [**macsec**]
4. **key** *hex-string*
5. **cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**}
6. **key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
7. **lifetime local** {{*day month year* **duration** *seconds*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *key-chain-name* [**macsec**] **Example:** `Device(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode
Step 4	**key** *hex-string* **Example:** `Device(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode. **Note** From Cisco IOS XE Everest Release 16.6.1 onwards, the Connectivity Association Key name (CKN) uses exactly the same string, which is configured as the hex-string for the key. For more information about this behavior change, see the section titled "MKA-PSK: CKN Behavior Change" after this task.
Step 5	**cryptographic-algorithm** {**gcm-aes-128**	**gcm-aes-256**} **Example:** `Device(config-keychain-key)# cryptographic-algorithm gcm-aes-128`	Set cryptographic authentication algorithm.
Step 6	**key-string** {[**0**	**6**] *pwd-string*	**7**	*pwd-string*} **Example:** `Device(config-keychain-key)# key-string 0 pwd`	Sets the password for a key string.

	Command or Action	Purpose
Step 7	**lifetime local** {{*day month year* **duration** *seconds*} **Example:** `Device(config-keychain-key)# lifetime local 16:00:00 Nov 9 2014 duration 6000`	Sets the lifetime for a key string. The range you can specify for the duration is between 1 and 864000 seconds.
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Returns to privileged EXEC mode.

Example for Connectivity Association Key (CAK) Rekey

CAK rekey will happen in the following cases:

- When moving from Key 01 to Key 02 within the Key Chain K1.
- When moving from one Key Chain K1 to another Key Chain K2.

Note: It is recommended to configure keys such that there is an overlap between the lifetime of the keys so that CAK rekey is successful and there is a seamless transition between the Keys/CA (without any traffic loss or session restart)

```
Device# show key chain k1
Key-chain k1:
    MacSEC key chain
    key 01 — text "c890433a1e05ef42d723a6b58af8fdbf7a25f42b3cda6a5eeb5ae4bf3a0a679f"
        lifetime (00:00:00 UTC Oct 29 2014) - (12:10:00 UTC Oct 29 2014)
    key 02 — text "14d9167d538819405c0ff78c655141ed4b3c7242562c0fb0f7a56f780bf29e52"
        lifetime (12:00:00 UTC Oct 29 2014) - (18:05:00 UTC Oct 29 2014)
    key 03 — text "88d971cb19d9f2598ad76edc562ade2e7e91e3ed70524f5c3c4d8d9599d0670e"
        lifetime (18:00:00 UTC Oct 29 2014) - (18:10:00 UTC Oct 29 2014)
    key 04 — text "75474bce819b49ad7e5bd06236bc0c944c69892f71e942e2f9812b7d3a7b2a5f"
        lifetime (18:10:00 UTC Oct 29 2014) - (infinite)

!In this case, Key 01, 02, 03 have overlapping time, but not key 04. Here is the sequence,
 how this works:
  @00:00:00 - A new MKA session is Secured with key 01
  @12:00:00 - CAK Rekey triggers with key 02 and upon success goes to Secured state
  @18:00:00 - CAK Rekey triggers with key 03 and upon success goes to Secured state
  @18:10:00 - Key 03 dies, hence MKA sesion using this key is brought down
  @18:10:00 - Key 04 becomes active and a new MKA session is triggered with this key. Upon
 success, session will be Secured and UP for infinite time.
```

MKA-PSK: CKN Behavior Change

From Cisco IOS XE Everest Release 16.6.1 onwards, for MKA-PSK sessions, instead of fixed 32 bytes, the Connectivity Association Key name (CKN) uses exactly the same string as the CKN, which is configured as the hex-string for the key.

Example Configuration:

```
configure terminal
key chain abc macsec
  key 11
    cryptographic-algorithm aes-128-cmac
    key-string 12345678901234567890123456789013
    lifetime local 12:21:00 Sep 9 2015 infinite

 end
```

For the above example, the following will be the **show** command output for the **show mka session** command:

```
Device# show mka session

           Total MKA Sessions....... 1
           Secured Sessions... 1
           Pending Sessions... 0

=============================================================================================

          Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server

          Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN


=============================================================================================

          Et0/0          aabb.cc00.6600/0002     icv            NO                NO

          2              aabb.cc00.6500/0002 1               Secured           11   *Note
 that the CKN key-string is exactly the same that has been configured for the key as
hex-string.*
```

In the case of interoperability between two images -- one having the CKN behavior change, and one without the CKN behavior change, then the hex-string for the key must be a 64-character hex-string with zero padded to work on a device that has an image with the CKN behavior change. See the example below:

Configuration without CKN key-string behavior change:

```
config t
  key chain abc macsec
   key 11
   cryptographic-algorithm aes-128-cmac
   key-string 12345678901234567890123456789013
   lifetime local 12:21:00 Sep 9 2015 infinite
```

Configuration with CKN key-string behavior change:

```
config t
  key chain abc macsec
   key 11000000000000000000000000000000000000000000000000000000000000000000
   cryptographic-algorithm aes-128-cmac
   key-string 12345678901234567890123456789013
   lifetime local 12:21:00 Sep 9 2015 infinite
```

Configuring an Option to Change the EAPoL Ethernet Type

-

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **eapol** *eth-type*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.
Step 4	**eapol** *eth-type* **Example:** `Device(config-if)# eapol eth-type 0xB860`	Configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface. **Note** From Cisco IOS Release XE 3.17, the **macsec eth-type** command has been replaced by the **eapol eth-type** command.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring Destination MAC Address on Interface and Sub-interface

Perform the following task to configure destination MAC address on the Interface or Subinterface. The destination MAC could be the MAC of the peer or a multicast MAC address. When the **eapol destination-address** command is configured on the main interface, it is applied to any subinterfaces on that interface. However, if the **eapol destination-address** command is configured on the subinterface, that takes take precedence over the command on the main interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **eapol destination-address** [*MAC-Address* | [**bridge-group-address** | **broadcast-address** | **lldp-multicast-address**]
5. **eapol destination-address bridge-group-address**
6. **eapol destination-address broadcast-address**
7. **eapol destination-address lldp-multicast-address**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.
Step 4	**eapol destination-address** [*MAC-Address*	[**bridge-group-address**	**broadcast-address**	**lldp-multicast-address**] **Example:** `Device(config-if)# eapol destination-address 0018.b967.3cd0`	Configures an Extensible Authentication Protocol over LAN (EAPoL) destination MAC address on the interface.
Step 5	**eapol destination-address bridge-group-address** **Example:** `Device(config-if)# eapol destination-address bridge-group-address`	Sets the destination address as a bridge group.
Step 6	**eapol destination-address broadcast-address** **Example:** `Device(config-if)# eapol destination-address broadcast-address`	Sets the destination address as a broadcast address.
Step 7	**eapol destination-address lldp-multicast-address** **Example:** `Device(config-if)# eapol destination-address lldp-multicast-address`	Sets the destination address as a LLDP multicast address.

	Command or Action	Purpose
Step 8	**end** **Example:** `DeviceDevice(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for WAN MACsec and MKA

Example: Point-to-point, CE to CE Connectivity Using EPL Service

The following is the sample configuration for point-to-point, Customer Edge to Customer Edge connectivity using Ethernet Private Line (EPL) using port-based service.

```
!Customer Edge 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!Customer Edge 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
```

Example: Point-to-point, Hub and Spoke Connectivity using EVPL Service

The following is sample configuration for point-to-point, hub and spoke connectivity using Ethernet Virtual Private Line (EVPL) Service in VLAN mode.

```
!CE 1
key chain k1 macsec*
 key 01
 key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
interface GigabitEthernet0/0/4.2
  encapsulation dot1Q 20
  ip address 10.3.2.1 255.255.255.0
  mka pre-shared-key key-chain k1*
```

```
  macsec*

!CE 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
   ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE 3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
```

Note All commands with asterix (*) are mandatory commands.

Example: Point-to-point, Hub and Spoke Connectivity with MACsec and non-MACsec Spokes

The following is sample output of point-to-point, Hub and Spoke Connectivity with MACsec and non-MACsec spokes.

```
!CE1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec access-control should-secure*
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
interface GigabitEthernet0/0/4.2
  encapsulation dot1Q 20
  ip address 10.3.2.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
interface GigabitEthernet0/0/4.3
  encapsulation dot1Q 30
  ip address 10.3.3.1 255.255.255.0
```

```
!CE2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec access-control should-secure*
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
   ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 20
  ip address 10.3.2.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*


!CE4
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 30
  ip address 10.3.3.2 255.255.255.0
```

Example: Multipoint-to-multipoint, Hub and Spoke connectivity using EP-LAN Service

The following example shows sample configuration multipoint-to-multipoint, hub and Spoke connectivity using Ethernet Private LAN (EP-LAN) Service in port mode.

```
!CE 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/4
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  mka policy p1
  macsec*

!CE 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/4
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
```

```
  mka policy p1
  macsec*

!CE 3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/4
  ip address 10.3.1.3 255.255.255.0
  mka pre-shared-key key-chain k1*
  mka policy p1
  macsec*
```

Example: Multipoint-to-multipoint, Hub and Spoke Connectivity Using EVP-LAN Service

The following is sample configuration for multipoint-to-multipoint, hub and spoke connectivity using Ethernet Virtual Private LAN (EVP-LAN) Service in VLAN mode:

```
!CE 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
  eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
  eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE 3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
  eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
```

```
  ip address 10.3.1.3 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
```

Example: Performing Maintenance Tasks Without Impacting Traffic

The following are sample configurations of performance maintenance tasks that do not impact traffic:

Changing a Pre-Shared Key (CAK Rollover)

The following is sample configuration for changing a pre-shared key:

Note Keys can be configured to automatically roll over to the next key by configuring a lifetime on both routers.

```
!From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012

!To
key chain k1 macsec*
  key 01
   key-string 12345678901234567890123456789012
   lifetime local 10:30:00 Oct 30 2014 11:30:00 Oct 30 2014
  key 02
   key-string 11145678901234567890123456789012
```

Changing a Key Chain (Keychain Rollover)

The following is the sample configuration for changing a key chain—Keychain Rollover

```
! From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k1

! To
key chain k1 macsec
  key 01
   key-string 12345678901234567890123456789012
key chain k2 macsec
  key 02
   key-string abcdef0987654321abcdef0987654321
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k2
```

Note The defined key ID, under any key chain, should be a unique value on the device.

A router can be become a key server by configuring a lower priority than other peer routers that participate in the same session. Configure a key server priority so that the key server selection is

deterministic. For example, in a Hub and Spoke scenario, the most ideal place for a key server is the Hub site router.

```
!Hub Site (Key Server):
mka policy p1
key-server priority 0
!0 is the default.

interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k1
mka policy p1

!Spoke Sites (non-Key Servers):
mka policy p1
key-server priority 1

interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k1
mka policy p1
```

The following is sample configuration for changing Cipher Suite to encrypt data traffic:

```
mka policy p1
 macsec-cipher-suite gcm-aes-128
interface GigabitEthernet0/0/1.10
 mka policy p1

!Alternate configuration

mka policy p1
 macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/1.10
 mka policy p1
```

```
key chain k3 macsec
 key 01
   key-string abcdef0987654321abcdef0987654321
   cryptographic-algorithm aes-128-cmac
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k3

!Alternate configuration:

key chain k3 macsec
 key 01
   key-string abcdef0987654321abcdef0987654321
   cryptographic-algorithm aes-256-cmac
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k3
```

EAPOL Destination MAC address can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the subinterfaces, if configured at the physical interface level. To override the inherited value, configure the MAC address at the subinterface mode. Default EAPOL destination MAC address is 01:80:c2:00:00:03.

```
interface TenGigabitEthernet0/0/0
eapol destination-address <H.H.H>

!Alternate configuration

interface TenGigabitEthernet0/0/0
 bridge-group-address

!Alternate configuration
```

```
interface TenGigabitEthernet0/0/0
 lldp-multicast-address>

mka policy p1
 confidentiality-offset 30
interface GigabitEthernet0/0/1.10
 mka policy p1
```

Example: Performing Maintenance Tasks—Traffic Impacting

Changing a Replay Protection Window Size

Replay protection window can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the sub interfaces if configured at the physical interface level. If you need to override the inherited value, configure it at the subinterface mode. The default replay protection window size is 64.

```
interface TenGigabitEthernet0/0/0
macsec replay-protection window-size 10

interface TenGigabitEthernet0/0/0.10
macsec replay-protection window-size 5
```

Enabling or Disabling VLAN (dot1q) Tag in the Clear Option

The **macsec dot1q-in-clear** command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

```
interface GigabitEthernet0/0/1
macsec dot1q-in-clear 1
```

The **macsec access-control** [**must-secure** | **should-secure**] command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

```
interface GigabitEthernet0/0/1
macsec access-control must-secure|should-secure
```

Example: Port-Channel Configuration with MACsec

The following is the sample configuration for port-channel configuration with MACsec on two seperate interfaces of a link bundle.

Note Before enabling or removing MACsec configuration from port channels, ensure that all the interfaces are shutdown.

```
key chain kc1 macsec
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac

key chain kc2 macsec
  key 02
```

```
  key-string 12345678901234567890123456789013
  cryptographic-algorithm aes-128-cmac

mka policy policy1
  macsec-cipher-suite gcm-aes-256

!Port-Channel Configuration

 interface Port-channel2
  mtu 9216
  ip mtu 9184
  ip address 10.3.1.3 255.255.255.0
  load-interval 30
  bfd interval 750 min_rx 750 multiplier 5
  lacp min-bundle 2
  no shut
  exit

!Member link configuration 1

 interface TenGigabitEthernet0/1/1
  no shut
  mtu 9216
  no ip address
  ip mtu 9184
  load-interval 30
  cdp enable
  no cdp tlv app
  mka policy policy1
  mka pre-shared-key key-chain kc1
  macsec
  lacp rate fast
  channel-group 2 mode active

!Member link configuration 2

  interface TenGigabitEthernet0/1/2
  no shut
  mtu 9216
  no ip address
  ip mtu 9184
  load-interval 30
  cdp enable
  no cdp tlv app
  mka policy policy1
  mka pre-shared-key key-chain kc2
  macsec
  lacp rate fast
  channel-group 2 mode active
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
IEEE 802.1AE-2006	*Media Access Control (MAC) Security*
IEEE 802.1X-2010	*Port-Based Network Access Control*
IEEE 802.1AEbw-2013	*Media Access Control (MAC) Security* (Amendment to IEEE 802.1AE-2006)—Extended Packet Numbering (XPN)
IEEE 802.1Xbx-2014	*Port-Based Network Access Control* (Amendment to IEEE 802.1X-2010)
RFC 4493	*The AES-CMAC Algorithm*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 105

MACsec Smart Licensing

MACsec Smart Licensing Overview

This chapter provides an overview of MACsec Smart Licensing. Smart Licensing feature is a standardized licensing platform that simplifies the Cisco software experience and helps you to understand how Cisco software is used across your network. Smart Licensing is the next generation licensing platform for all Cisco software products. MACsec licensing allows you to enable CSL permanent and Smart Licensing on Cisco ASR 1000 platforms.

Feature Information for MACsec Smart Licensing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 143: Feature Information for MACsec Smart Licensing

Feature Name	Releases	Feature Information
MACsec and DLC Support	Cisco IOS XE Fuji 16.9.1	Smart Licensing feature is a standardized licensing platform that simplifies the Cisco software experience and helps you to understand how Cisco software is used across your network. Smart Licensing is the next generation licensing platform for all Cisco software products. No commands were introduced or modified by this feature.

Information about MACsec Smart Licensing

Effective with Cisco IOS XE Fuji Release 16.9.1, MACsec Smart Licensing (SL) is supported on the following platforms:

Ports	License Feature	License PID	Supported Platform		
			MIP-100 (RP2/RP3)	ASR1001-HX	ASR1002-HX
BUILT-IN 1 GE ports	MACSEC1G	FLSA1-MACSEC1G	N/A	Yes	Yes
BUILT-IN 10 GE ports	MACSEC10G	FLSA1-MACSEC10G	N/A	Yes	Yes
EPA-18X1GE	MACSEC1G	FLSA1-MACSEC1G	Yes	N/A	Yes
EPA-10X10GE	MACSEC10G	FLSA1-MACSEC10G	Yes	N/A	Yes
EPA-1X40GE	MACSEC40G	FLSA1-MACSEC40G	Yes	N/A	Yes
EPA-2X40GE	MACSEC40G	FLSA1-MACSEC40G	Yes	N/A	Yes
EPA-QSFP-1X100GE	MACSEC100G	FLSA1-MACSEC100G	Yes	N/A	Yes

MACsec licenses are available for each port and are applicable only for physical ports (sub-interfaces do not require additional license). Device Led Conversion (DLC) support is available for MACsec port licenses to ensure that your paper licenses are added to smart account.

The Device-led conversion allows license migration from Classic to Smart license automatically for licences that are on the devices. The devices needs to be registered in Cisco Smart Software Manager (SSM) for automatic conversion to smart license.

Note

- According to earlier releases, ASR1001-X built-in continues with the IPsec license which acts as MACsec license.
- MACsec license is not supported for EPA-1X100GE and EPA-CPAK-2X40GE.
- CSL – EvalRTU license is not available for MACsec licenses.

One unit of MACsec license is used when a port containing MACsec configuration is unshut or when the configuration is applied on an unshut port.

One unit of MACsec license is released when a port containing MACsec configuration is shut or when the configuration is removed from an unshut port.

Deployment and Migration Examples

MACSec support is available in Cisco Software License (CSL) and Smart License (SL) modes from Cisco IOS XE Fuji 16.9.1. However, for releases after 16.9.1, MACSec will support only Smart License.

The following scenarios explain how an existing router is deployed and migrated to Cisco IOS XE Fuji 16.9.1:

Upgrade in CSL Mode when a permanent license is installed

If MACsec permanent licenses are installed on the device before upgrading (prior to Cisco IOS XE Fuji 16.9.1 release), then these licenses are used after the upgrade.

- Before the upgrade, assume the following:
  - Router is operating on a release prior to Cisco IOS XE Fuji 16.9.1
  - MACsec is configured on four un-shut 1G interfaces
  - Four MACSEC1G permanent licenses are installed
- After the upgrade, four MACSEC1G licenses are used.

Upgrade in CSL Mode when a permanent license is not installed

When MACsec is configured on unshut ports, ideally EvalRTU licenses should be used after the upgrade. Since EvalRTU support is not available, the license request is skipped and a warning message is displayed. For example:

%IOSXE_LICENSE_POLICY_MANAGER-4-INSUF_PERM_LIC: 0/0/0: Insufficient MACSEC40G permanent license, skipping license request assuming customer has honour license

- Before the upgrade, assume the following:
  - Router is operating on a release prior to Cisco IOS XE Fuji 16.9.1
  - MACsec is configured on four un-shut 1G interfaces
- After the upgrade
  - No MACsec license is used

- Warning message is displayed
- If you install four permanent licenses at a later point of time, then these licenses are used immediately

Migration to SL Mode

To avoid **Out of Compliance** scenario, all Product Activation Keys (PAK) and non-PAK licenses should be added to customer's virtual CSSM account.

The Device Led Conversion (DLC) feature migrates licenses to Smart Account. For DLC to work properly, all licenses should be enabled in CSL mode before migrating to SL mode.

Perform the following steps to migrate to SL Mode:

- Upgrade from releases prior to Cisco IOS XE 16.9.1 to Cisco IOS XE 16.9.1
  1. Upgrade to Cisco IOS XE Fuji 16.9.1 in CSL mode
  2. Migrate to SL mode and trigger DLC
- Upgrade from releases prior to Cisco IOS XE Fuji 16.9.1 to later releases
  1. Upgrade to Cisco IOS XE Fuji 16.9.1 in CSL mode
  2. Migrate to SL mode and trigger DLC
  3. Upgrade to releases later than Cisco IOS XE Fuji 16.9.1

CHAPTER 106

Certificate-based MACsec Encryption

The Certificate-based MACsec Encryption feature uses 802.1X port-based authentication with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) to carry Certificates for router ports where MACsec encryption is required. EAP-TLS mechanism is used to mutually authenticate and get the Primary Session Key from which the Connectivity Association Key (CAK) is derived for the MACsec Key Agreement (MKA) protocol.

Certificate-based MACsec encryption can be done using either remote authentication or local authentication.

Feature Information for Certificate-based MACsec Encryption

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 144: Feature Information for Certificate-based MACsec Encryption

Feature Name	Releases	Feature Information
Certificate-based MACsec Encryption	Cisco IOS XE Everest Release 16.6.1	The Certificate-based MACsec Encryption feature uses 802.1X port-based authentication with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) to carry Certificates for router ports where MACsec encryption is required. EAP-TLS mechanism is used to do the mutual authentication and to get the Primary Session Key from which the Connectivity Association Key (CAK) is derived for the MACsec Key Agreement (MKA) protocol.

Prerequisites for Certificate-based MACsec Encryption

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0. Refer to the *Cisco Identity Services Engine Administrator Guide, Release 2.3*.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.
- Ensure that 802.1x authentication and AAA are configured on your device.

Restrictions for Certificate-based MACsec Encryption

- MKA is not supported on port-channels.
- High Availability for MKA is not supported.
- Certificate-based MACsec encryption on sub-interfaces is not supported.

Information About Certificate-based MACsec Encryption

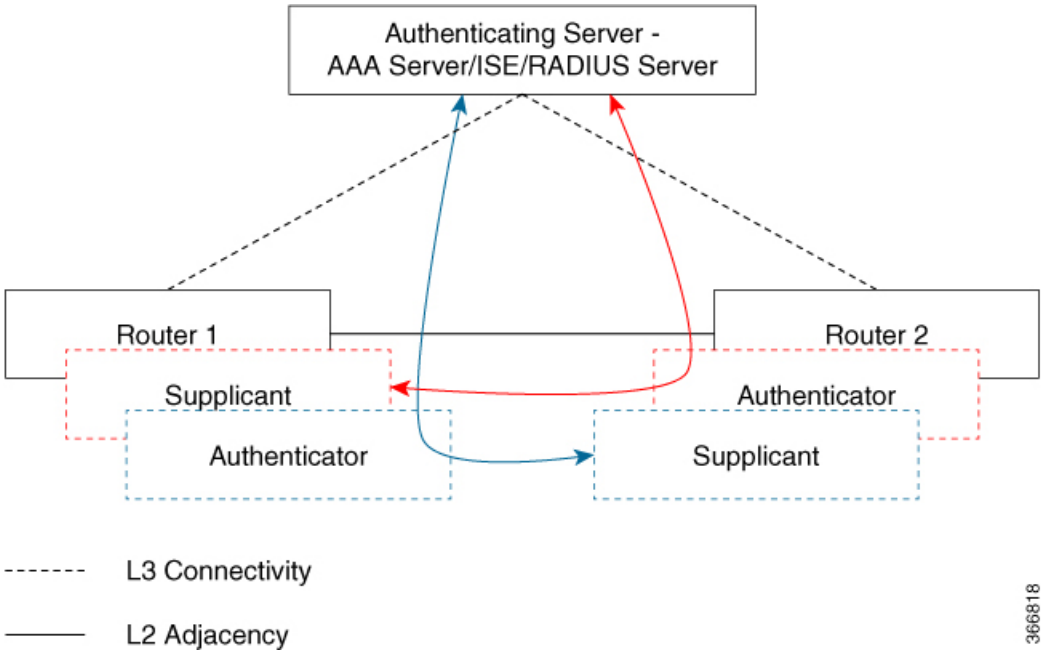
MKA MACsec is supported on router-to-router links. Using IEEE 802.1X Port-based Authentication with Extensible Authentication Protocol (EAP-TLS), you can configure MKA MACsec between device ports. EAP-TLS allows mutual authentication and obtains an primary session key from which the connectivity association key (CAK) is derived for MKA protocol. Device certificates are carried, using EAP-TLS, for authentication to the AAA server.

Call Flow for Certificate-based MACsec Encryption using Remote Authentication

Suppllicants are unauthorized devices that try to gain access to the network. Authenticators are devices that control the physical access to the network based on the authentication status of the supplicant.

As shown in the following diagram, the devices are connected directly. The router acts as both EAP Supplicant and Authenticator on the port.

The figure below depicts two EAP call flows (with separate EAP-Session ID) on the router. The red flow depicts Router 1 as supplicant and Router 2 as authenticator and the blue flow is vice-versa.



When the interface is configured for 802.1x role as both, The authentication manager on a router creates a session with two EAP session (blue and red with separate EAP session ID) flows with supplicant as well as an authenticator role and both trigger EAP-TLS mutual authentication with the remote authenticating server (AAA server/ISE/RADIUS).

After mutual authentication, the MSK of the flow corresponding to the router with the higher MAC address and role as authenticator is picked to derive the CAK.

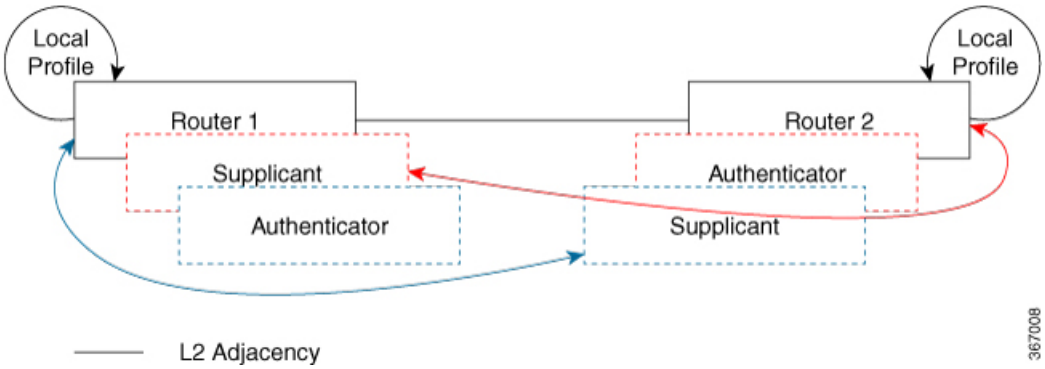
In the diagram above, if Router 1 MAC address is less than Router 2, then the primary session key (PSK) obtained from the EAP session (blue flow) is used as EAP-PSK for the MKA (Router 1 acts as authenticator and Router 2 as supplicant). This ensures that Router 1 acts as MKA Key Server and Router 2 is the Non-Key Server.

If the Router 2 MAC Address is less than Router 1 then the PSK obtained from the EAP session (red flow) is used (by both routers) as EAP-PSK for the MKA to derive the CAK.

Call Flow for Certificate-based MACsec Encryption using Local Authentication

As shown in the following diagram, the devices are connected directly. The router acts as both EAP Supplicant and Authenticator on the port.

The figure below depicts two EAP call flows (with separate EAP-Session ID) on the router. The red flow depicts Router 1 as supplicant and Router 2 as authenticator and the blue flow is vice-versa.



When the interface is configured for 802.1x role as both, The authentication manager on a router creates a session with two EAP session (blue and red with separate EAP session ID) flows with supplicant as well as an authenticator role and both trigger EAP-TLS mutual authentication with the local authenticating server.

After mutual authentication, the PSK of the flow corresponding to the router with the higher MAC address and role as authenticator is picked to derive the CAK.

In the diagram above, if Router 1 MAC address is less than Router 2, then the primary session key (PSK) obtained from the EAP session (blue flow) is used as EAP-PSK for the MKA (Router 1 acts as authenticator and Router 2 as supplicant). This ensures that Router 1 acts as MKA Key Server and Router 2 is the Non-Key Server.

If the Router 2 MAC Address is less than Router 1 then the PSK obtained from the EAP session (red flow) is used (by both routers) as EAP-PSK for the MKA to derive the CAK.

Configuring Certificate-based MACsec Encryption using Remote Authentication

To configure MACsec with MKA on point-to-point links, perform these tasks:

Configuring Certificate Enrollment

Generating Key Pairs

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto key generate rsa label** *label name* **general-keys modulus** *size*	Generates a RSA key pair for signing and encryption. You can also assign a label to each key pair using the label keyword. The label is referenced by the trustpoint that uses the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>. If you do not use additional keywords this command generates one general purpose RSA key pair. If the modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the modulus keyword.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show authentication session interface** *interface-id*	Verifies the authorized session security status.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Enrollment using SCEP

Simple Certificate Enrollment Protocol (SCEP) is a Cisco-developed enrollment protocol that uses HTTP to communicate with the certificate authority (CA) or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**rsakeypair** *label*	Specifies which key pair to associate with the certificate. **Note** The **rsakeypair** name must match the trust-point name.

	Command or Action	Purpose
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**auto-enroll** *percent* **regenerate**	Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA. If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration. By default, only the Domain Name System (DNS) name of the device is included in the certificate. Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: "! RSA key pair associated with trustpoint is exportable." It is recommended that a new key pair be generated for security reasons.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**exit**	Exits global configuration mode.
Step 12	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.

Configuring Enrollment Manually

If your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform the following task to set up manual certificate enrollment:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**rsakeypair** *label*	Specifies which key pair to associate with the certificate.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**exit**	Exits Global Configuration mode.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**crypto pki enroll** *name*	Generates certificate request and displays the request for copying and pasting into the certificate server. Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate**	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch.

	Command or Action	Purpose
		Note Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 13	**exit**	Exits Global Configuration mode.
Step 14	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.
Step 15	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Enabling 802.1x Authentication and Configuring AAA

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**dot1x system-auth-control**	Enables 802.1X on your device.
Step 5	**radius server** *name*	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	**address** *ip-address* **auth-port** *port-number* **acct-port** *port-number*	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 7	**automate-tester username** *username*	Enables the automated testing feature for the RADIUS server. With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks for a RADIUS response from the server. A success message is not necessary - a failed authentication suffices, because it shows that the server is alive.
Step 8	**key** *string*	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.

	Command or Action	Purpose
Step 9	**radius-server deadtime** *minutes*	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
Step 10	**exit**	Returns to global configuration mode.
Step 11	**aaa group server radius** *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
Step 12	**server** *name*	Assigns the RADIUS server name.
Step 13	**exit**	Returns to global configuration mode.
Step 14	**aaa authentication dot1x default group** *group-name*	Sets the default authentication server group for IEEE 802.1x.
Step 15	**aaa authorization network default group** *group-name*	Sets the network authorization default group.

Configuring EAP-TLS Profile and 802.1x Credentials

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**eap profile** *profile-name*	Configures EAP profile and enters EAP profile configuration mode.
Step 4	**method tls**	Enables EAP-TLS method on the device.
Step 5	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**dot1x credentials** *profile-name*	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	**username** *username*	Sets the authentication user ID.
Step 9	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 10	**end**	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 4	**macsec**	Enables MACsec on the interface.
Step 5	**authentication periodic**	Enables reauthentication for this port.
Step 6	**authentication timer reauthenticate interval**	Sets the reauthentication interval.
Step 7	**access-session host-mode multi-domain**	Allows hosts to gain access to the interface.
Step 8	**access-session closed**	Prevents preauthentication access on the interface.
Step 9	**access-session port-control auto**	Sets the authorization state of a port.
Step 10	**dot1x pae both**	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	**dot1x credentials profile**	Assigns a 802.1x credentials profile to the interface.
Step 12	**dot1x supplicant eap profile** *name*	Assigns the EAP-TLS profile to the interface.
Step 13	**service-policy type control subscriber** *control-policy name*	Applies a subscriber control policy to the interface.
Step 14	**exit**	Returns to privileged EXEC mode.
Step 15	**show macsec interface**	Displays MACsec details for the interface.
Step 16	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Certificate-based MACsec Encryption using Local Authentication

To configure MACsec with MKA on point-to-point links, perform these tasks:

Configuring the EAP Credentials using Local Authentication

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**aaa local authentication default authorization default**	Sets the default local authentication and default local authorization method.
Step 5	**aaa authentication dot1x default local**	Sets the default local username authentication list for IEEE 802.1x.
Step 6	**aaa authorization network default local**	Sets an authorization method list for local user.
Step 7	**aaa authorization credential-download default local**	Sets an authorization method list for use of local credentials.
Step 8	**exit**	Returns to privileged EXEC mode.

Configuring the Local EAP-TLS Authentication and Authorization Profile

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**dot1x credentials** *profile-name*	Configures the dot1x credentials profile and enters dot1x credentials configuration mode.
Step 5	**username** *name* **password** *password*	Sets the authentication user ID and password.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**aaa attribute list** *list-name*	(Optional) Sets the AAA attribute list definition and enters attribute list configuration mode.
Step 8	**aaa attribute type linksec-policy must-secure**	(Optional) Specifies the AAA attribute type.
Step 9	**exit**	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	**username** *name* **aaa attribute list** *name*	(Optional) Specifies the AAA attribute list for the user ID.
Step 11	**end**	Returns to privileged EXEC mode.

Configuring Enrollment using SCEP

Simple Certificate Enrollment Protocol (SCEP) is a Cisco-developed enrollment protocol that uses HTTP to communicate with the certificate authority (CA) or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**rsakeypair** *label*	Specifies which key pair to associate with the certificate. **Note** The **rsakeypair** name must match the trust-point name.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**auto-enroll** *percent* **regenerate**	Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA. If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration.

	Command or Action	Purpose
		By default, only the Domain Name System (DNS) name of the device is included in the certificate. Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: "! RSA key pair associated with trustpoint is exportable." It is recommended that a new key pair be generated for security reasons.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**exit**	Exits global configuration mode.
Step 12	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.

Configuring Enrollment Manually

If your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform the following task to set up manual certificate enrollment:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.

	Command or Action	Purpose
Step 5	**rsakeypair** *label*	Specifies which key pair to associate with the certificate.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**exit**	Exits Global Configuration mode.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**crypto pki enroll** *name*	Generates certificate request and displays the request for copying and pasting into the certificate server. Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate**	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 13	**exit**	Exits Global Configuration mode.
Step 14	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.

	Command or Action	Purpose
Step 15	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring EAP-TLS Profile and 802.1x Credentials

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**eap profile** *profile-name*	Configures EAP profile and enters EAP profile configuration mode.
Step 4	**method tls**	Enables EAP-TLS method on the device.
Step 5	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**dot1x credentials** *profile-name*	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	**username** *username*	Sets the authentication user ID.
Step 9	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 10	**end**	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.

	Command or Action	Purpose
Step 4	**macsec**	Enables MACsec on the interface.
Step 5	**authentication periodic**	Enables reauthentication for this port.
Step 6	**authentication timer reauthenticate interval**	Sets the reauthentication interval.
Step 7	**access-session host-mode multi-domain**	Allows hosts to gain access to the interface.
Step 8	**access-session closed**	Prevents preauthentication access on the interface.
Step 9	**access-session port-control auto**	Sets the authorization state of a port.
Step 10	**dot1x pae both**	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	**dot1x credentials profile**	Assigns a 802.1x credentials profile to the interface.
Step 12	**dot1x authenticator eap profile** *name*	Assigns the EAP-TLS authenticator profile to the interface.
Step 13	**dot1x supplicant eap profile** *name*	Assigns the EAP-TLS supplicant profile to the interface.
Step 14	**service-policy type control subscriber** *control-policy name*	Applies a subscriber control policy to the interface.
Step 15	**exit**	Returns to privileged EXEC mode.
Step 16	**show macsec interface**	Displays MACsec details for the interface.
Step 17	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Verifying Certificate-based MACsec Encryption

Use the following **show** commands to verify the configuration of certificate-based MACsec encryption. Given below are the sample outputs of the **show** comamnds.

The **show mka sessions** command displays a summary of active MACsec Key Agreement (MKA) Protocol sessions.

```
Device# show mka sessions

  Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

=============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server

Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN

=============================================================================================
Te0/1/3        74a2.e625.4413/0013 *DEFAULT POLICY* NO                YES
```

```
19             74a2.e625.4c22/0012 1                Secured
10000000000000000000000000000000000000000000000000000000000000000000
```

The **show macsec status interface** *interface-id* displays MACsec status information for the given interface.

```
Device# show macsec status interface te0/1/2

  Capabilities:
  Ciphers Supported:        GCM-AES-128 GCM-AES-256
  Cipher:                   GCM-AES-128
  Confidentiality Offset:   0
  Replay Window:            64
  Delay Protect Enable:     FALSE
  Access Control:           must-secure

Transmit SC:
  SCI:                      74A2E6254C220012
  Transmitting:             TRUE
Transmit SA:
  Next PN:                  412
  Delay Protect AN/nextPN:  99/0

Receive SC:
  SCI:                      74A2E62544130013
  Receiving:                TRUE
Receive SA:
  Next PN:                  64
  AN:                       0
  Delay Protect AN/LPN:     0/0
```

The **show access-session interface** *interface-id* **details** displays detailed information about the access session for the given interface.

```
Device# show access-session interface te1/0/1 details

   Interface:  TenGigabitEthernet1/0/1
             IIF-ID:  0x17298FCD
         MAC Address:  f8a5.c592.13e4
        IPv6 Address:  Unknown
        IPv4 Address:  Unknown
           User-Name:  DOT1XCRED
              Status:  Authorized
              Domain:  DATA
      Oper host mode:  multi-host
    Oper control dir:  both
     Session timeout:  N/A
   Common Session ID:  000000000000000BB72E8AFA
     Acct Session ID:  Unknown
              Handle:  0xc3000001
      Current Policy:  MUSTS_1


Local Policies:
 Security Policy:  Must Secure
      Security Status:  Link Secured

Server Policies:


Method status list:
```

```
      Method           State
    dot1xSup           Authc Success
       dot1x           Authc Success
```

Configuration Examples for Certificate-based MACsec Encryption

Example: Enrolling the Certificate

```
Configure Crypto PKI Trustpoint:
  crypto pki trustpoint POLESTAR-IOS-CA
   enrollment terminal
   subject-name CN=ASR1000x1@polestar.com, C=IN, ST=KA, OU=ENG,O=Polestar
   revocation-check none
   rsakeypair mkaioscarsa
   storage nvram:
  !
Manual Installation of Root CA certificate:
 crypto pki authenticate POLESTAR-IOS-CA
```

Example: Enabling 802.1x Authentication and AAA Configuration

```
aaa new-model
dot1x system-auth-control
radius server ISE
 address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646
 automate-tester username dummy
 key dummy123
 radius-server deadtime 2
!
aaa group server radius ISEGRP
 server name ISE
!
aaa authentication dot1x default group ISEGRP
aaa authorization network default group ISEGRP
```

Example: Configuring EAP-TLS Profile and 802.1X Credentials

```
eap profile EAPTLS-PROF-IOSCA
 method tls
 pki-trustpoint POLESTAR-IOS-CA
!

dot1x credentials EAPTLSCRED-IOSCA
 username asr1000@polestar.company.com
 pki-trustpoint POLESTAR-IOS-CA
!
```

Example: Applying 802.1X, PKI, and MACsec Configuration on the Interface

```
interface TenGigabitEthernet0/1
 macsec network-link
 authentication periodic
 authentication timer reauthenticate <reauthentication interval>
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae both
 dot1x credentials EAPTLSCRED-IOSCA
 dot1x supplicant eap profile EAPTLS-PROF-IOSCA
 service-policy type control subscriber DOT1X_POLICY_RADIUS
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
IEEE 802.1AE-2006	*Media Access Control (MAC) Security*
IEEE 802.1X-2010	*Port-Based Network Access Control*
IEEE 802.1AEbw-2013	*Media Access Control (MAC) Security* (Amendment to IEEE 802.1AE-2006)—Extended Packet Numbering (XPN)
IEEE 802.1Xbx-2014	*Port-Based Network Access Control* (Amendment to IEEE 802.1X-2010)
RFC 4493	*The AES-CMAC Algorithm*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 107

MACsec as a Service-An Encryption Solution

This document describes how to deploy an encryption solution - Cisco MACsec as a Service, to secure network traffic using Cisco WAN MACsec and Ethernet Virtual Circuit (EVC). This solution provides Ethernet Virtual Circuit (EVC) support for Media Access Control security (MACsec) with MACsec Key Agreement (MKA) protocol. MACsec with MKA detects EVCs and enables the physical interface that matches the EVC criteria. With this functionality, users can transport layer2 traffic from multiple enterprises over a WAN link and independently secure their traffic with MACsec over EVC.

Feature Information for MACsec as a Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 145: Feature Information for MACsec as a Service

Feature Name	Releases	Feature Information
MACsec as a Service - Ethernet Virtual Circuit Support for MACsec and MKA	Cisco IOS XE Gibraltar 16.12.1a	This document describes how to deploy an encryption solution using Ethernet Virtual Circuit (EVC) support for MACsec with MACsec Key Agreement (MKA) protocol. MACsec with MKA detects EVCs and enables the physical interface that matches the EVC criteria. With this functionality, users can transport layer 2 traffic from multiple enterprises over a WAN link and independently secure their traffic with MACsec over EVC. In this release, the feature is supported only on Cisco ASR1000 Series Aggregation Services Routers. The following commands were introduced or modified: **mka pre-shared-key key-chain** *key-chain-name*, **mka policy** *policy-name*, **mka default-policy**, **macsec replay-protection window** *window size*, **eapol destination-address** *destination-address*{*bridge-group-address* | *broadcast-address* | *lldp-multicast-address* | *unicast mac-address*}, **eapol eth-type** *eth-type* .

Prerequisites for Ethernet Virtual Circuit Support for MACsec and MKA

- WAN MACsec requires a MACsec license. See the Table in Cisco ASR 1000 Series Ethernet Line Cards Datasheet
- Ensure that the Layer2 transparent Ethernet Services are available. The service provider network must provide a MACsec Layer2 Control Protocol transparency, such as, Extensible Authentication Protocol over LAN (EAPoL).

Restrictions for Ethernet Virtual Circuit Support for MACsec and MKA

- This feature is supported only on Cisco 1000 Series Aggregation Services Routers.
- This feature is supported from Cisco IOS XE Gibraltar 16.12.1a.
- Only dot1q based header is supported on EVC with MACsec.

  Number of MKA P2P sessions per port is 8 on 1 Gig and 32 on 10 Gig interfaces.
- If MACsec or MKA session is already configured on a physical interface or on a sub-interface, then you cannot configure MACsec with MKA session under the service instance or EVC mode on the same physical interface and vice versa.
- MACsec EVC is supported only with MKA PSK based sessions.

Information About Ethernet Virtual Circuit Support for MACsec and MKA

MACsec and MKA Overview

MACsec is an IEEE 802.1AE standard based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) protocol provides the required session keys and manages the required encryption keys. Only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

The 802.1AE encryption with MKA is supported on downlink ports for encryption between the routers or switches and host devices. MKA is the control plane for MACsec, which is defined in the IEEE standard 802.1X. MKA frames form part of the EAPoL frames. MACsec is the last mile in the packet processing path and encrypts all the traffic except the EAPoL frames.

For implementing WAN MACsec and MKA, verify that a basic Layer 2 Ethernet connectivity is established before attempting to enable MACsec. For more information, refer to the MACsec and MKA Overview section.

Cisco Ethernet Virtual Circuit

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service. It embodies the different parameters on which the service is being offered. In the Cisco EVC structure, the bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given router. Service instance is associated with a bridge domain (BD) based on the configuration.

An incoming frame can be classified as service instance based on the following criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Service instance also supports the alternative mapping criteria:
- Untagged—Mapping to all the frames lacking a 802.1Q or 802.1ad header
- Default—Mapping to all the frames

For more information on the EVC architecture, see "Configuring Ethernet Virtual Circuit" section on the in the Carrier Ethernet Configuration guide.

Ethernet Service Instance or Ethernet Flow Point

Ethernet Flow Point (EFP) is a transport-agnostic abstraction of an Ethernet service on an interface. It classifies frames from a same physical port to one of the multiple service instances associated with the port based on the user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MACsec Key Agreement (MKA) is used as the control protocol. MKA selects the cipher suite, which is used for encryption and exchanges the required keys and parameters between peers.

MKA uses Extensible Authentication Protocol over LAN (EAPoL) as the transport protocol to transmit MKA messages. By default, EAPoL uses a destination multicast MAC address of 01:80:c2:00:00:03 to multicast packets to multiple destinations. EAPoL is a standards-based protocol and other authentication mechanisms such as IEEE 802.1X also use the same protocol. Devices in the service provider cloud might consume this packet (based on the destination multicast MAC address), and try to process the EAPoL packet and eventually drop the packet. This causes MKA session to fail.

Use the **eapol destination-address** command to change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider. This ensures that the service provider tunnels the packet like any other data packet instead of consuming them.

Note The EAPoL destination address can be configured on either physical or on a subinterface level. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on the subinterface overrides the inherited value or policy for that subinterface.

Bridge Domain (BD) defines a broadcast domain internal to the platform and it allows decoupling broadcast domain from VLAN thus enables per-port VLAN significance. This removes the scalability limitations associated with a single per-box VLAN ID space. For more information on how EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP), refer to Bridge Domain Interface Encapsulation.

Benefits of MACsec and MKA with Ethernet Virtual Circuit

- Transport the Layer2 VLANs from multiple enterprise customers over a WAN link and independently secure their traffic using MACsec.

  Selective encryption of the LAN traffic over WAN using MACsec

For more information on the benefits of WAN MACsec and MKA Support, refer to the Benefits of WAN MACsec and MKA Support Enhancements section.

MACsec as a Service using Ethernet Virtual Circuit

The topologies below describe how to deploy Ethernet Virtual Circuit (EVC) with WAN MACsec in an EoMPLS network in a Point-to-Point and Point to Multi-Point scenarios. The traffic, which is encrypted, flows from CEs with CVLAN to the CE routers, and the CE routers in the network ensure that the data reaches their destination.

Figure 31: MKA and MACsec Topology with a single SVLAN

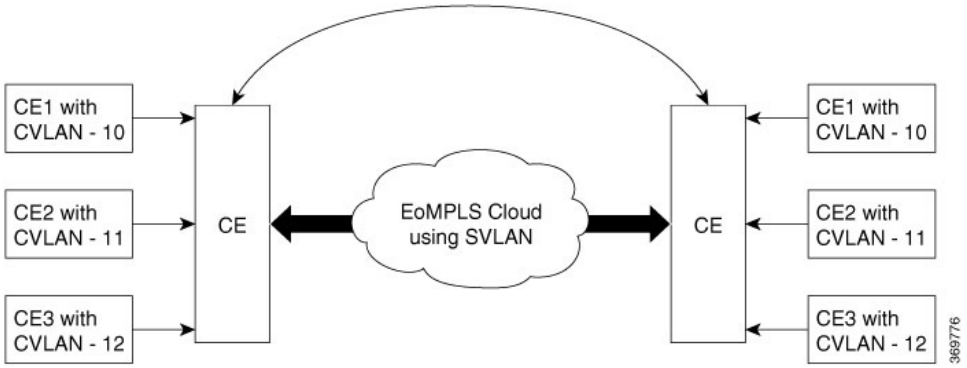
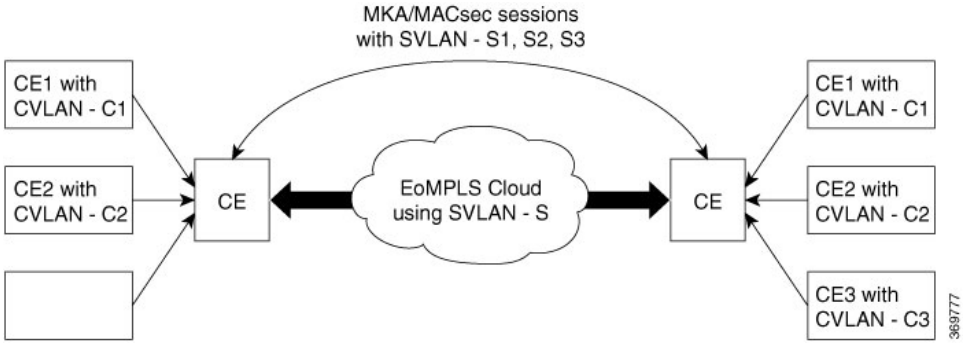


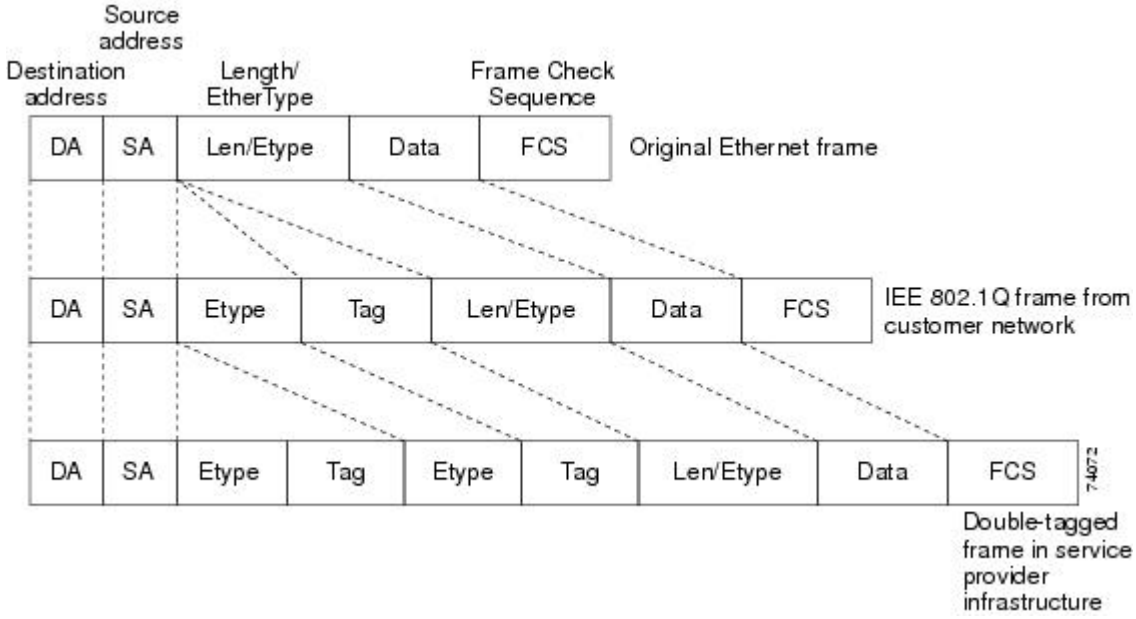
Figure 32: MKA and MACsec Topology with Multiple SVLANs



Cisco WAN MACsec, which supports EAPoL frames, not only encrypts the data, but, helps to seamlessly navigate across a diverse service provider network to securely connect all your remote sites.

In an EoMPLS network, you can connect multiple Layer 2 Ethernet networks at different locations. To enable connecting to different service providers over EoMPLS, WAN MACsec supports dot.1q tag in the clear, which helps connect to remote sites over public E-LINE or E-LAN services without disrupting the service provider network.

Figure 33: 802.1Q, and Double-Tagged Ethernet Packet Formats



Service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

When you use a service provider network to exchange data between networks, the EVC with MACsec helps to encrypt the data in transit. The dot.1q tag in clear opens a multitude of design options for securing complex networks. Using the EVCs, service providers can encapsulate packets that enter the service-provider network with multiple customer VLAN IDs (C-VLANs) and a single 0x8100 Ethertype VLAN tag with a service provider VLAN (S-VLAN). Within the service provider network, packets are switched based on the S-VLAN. When the packets egress the service provider network onto the customer network, the S-VLAN tag is decapsulated and the original customer packet is restored.

How to Configure Ethernet Virtual Circuit Support for MACsec and MKA

Configure Key Chain

To configure a key chain, perform the steps below:

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **key chain** *key-chain-name* **macsec**

Example:

```
Device(config)# Key chain keychain1 macsec
```

Configures a key chain and enters keychain configuration mode

Step 4 **key** *hex-string*

Example:

```
Device(config-keychain)# key 01
```

Configures a key and enters keychain key configuration mode.

Step 5 **cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**}

Example:

```
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
```

Set cryptographic authentication algorithm.

Step 6 **key-string** *pwd-string*}

Example:

```
Device(config-keychain-key)# key-string 12345678901234567890123456789013
```

Sets the password for a key string.

Step 7 **end**

Example:

```
Device(config-keychain-key)# end
```

Returns to privileged EXEC mode.

Configure MKA and MACsec on Interfaces

To configure MKA and MACsec on an interface, perform these steps:

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters the configuration mode

Step 3 **mka policy** *policy-name*

Example:

```
Device(config)# mka policy
```

Configures an MKA policy

Step 4 **mka pre-shared-key key-chain** *key-chain-name*

Example:

```
Device(config)# mka pre-shared-key key-chain 10
```

Configures an MKA pre-shared-key key-chain 10

Note The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.

Step 5 **macsec**

Configures MACsec for the EAPOL frame type.

Step 6 **macsec replay-protection window** *window-size*

Changes the replay window 10

Step 7 **end**

Returns to privileged EXEC mode.

Configure Ethernet Virtual Circuit on Ingress Port Facing Customer Edge

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Enters global configuration mode.

Step 3 **interface GigabitEthernet0/0/2**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 4 **service instance 10 Ethernet**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 5 **configure terminal**

Enters global configuration mode.

Step 6 **interface GigabitEthernet0/0/2**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 7 **encapsulation dot1q 10**

Step 8 **rewrite ingress tag push dot1q 20 symmetric**

Step 9 **bridge-domain** *number*

Step 10

```
interface GigabitEthernet0/0/2
   service instance 11 Ethernet
encapsulation dot1q 10
rewrite ingress tag push dot1q 20 symmetric
bridge-domain 21
interface GigabitEthernet0/0/2
   service instance 12 Ethernet
encapsulation dot1q 10
rewrite ingress tag push dot1q 20 symmetric
bridge-domain 22
```

Configure MACsec EVC on Egress Port Facing Service Provider Network

Step 1 **enable**

Step 2 **configure terminal**

Example:

```
interface tenGigabitEthernet0/1/1
    macsec dot1q-in-clear 1
    service instance 20 Ethernet
encapsulation dot1q 20
mka pre-shared-key key-chain kc1
macsec
bridge-domain 20
    service instance 21 Ethernet
encapsulation dot1q 21
mka pre-shared-key key-chain kc1
     macsec
     bridge-domain 21
    service instance 22 Ethernet
```

```
encapsulation dot1q 22
mka pre-shared-key key-chain kc1
macsec
bridge-domain 22
```

Verify Enablement of Pre-Shared-Key based on a Macsec and MKA session

SUMMARY STEPS

1. **enable**
2.

DETAILED STEPS

Step 1 **enable**

Step 2 **Example:**

```
show running-config | sec kc1
key chain kc1 macsec
 key 01
   cryptographic-algorithm aes-128-cmac
  key-string 12345678901234567890123456789012
 mka pre-shared-key key-chain kc1
  mka pre-shared key-chain kc1
```

The following is sample configuration for enabling Pre-Shared-Key (PSK) based MKA/Macsec session with default policy under service instance mode:

```
Device#show running-config interface gi0/0/0
Building configuration...
 …
 …
 …
Current configuration : 142 bytes
!
interface Ethernet0/0
  no ip address
  negotiation auto
  service instance 10 ethernet
    encapsulation dot1q 10
    rewrite ingress tag pop 1 symmetric
    mka pre-shared key-chain kc1
    macsec
    bridge-domain 100
 !
end
```

Configuration Examples for Ethernet Virtual Circuit Support for MACsec and MKA

Example: General Troubleshooting

Example: General Troubleshooting

Example: Show MKA Configured Command

Example: Show MKA Configured Command

Example: Show Statistics

```
MACsec statistics on an EFP: To validate MACsec Statistics on an EFP instance, use
show macsec statistics interface gi0/0/3 efp 10
------------------------------------------------------------
MACsec Statistics for Gi0/0/3.EFP10
SecY Counters
  Ingress Untag Pkts:       5
  Ingress No Tag Pkts:      63440
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
  Ingress No SCI Pkts:      0
  Ingress Overrun Pkts:     0
  Ingress Validated Octets: 0
  Ingress Decrypted Octets: 0
  Egress Untag Pkts:        0
  Egress Too Long Pkts:     0
  Egress Protected Octets:  0
  Egress Encrypted Octets:  0
Controlled Port Counters
  IF In Octets:             0
  IF In Packets:            0
  IF In Discard:            63440
  IF In Errors:             0
  IF Out Octets:            0
  IF Out Packets:           0
  IF Out Errors:            0
 Transmit SC Counters (SCI: 70708BBA4683000A)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       0
Transmit SA Counters (AN 2)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       0
Receive SA Counters (SCI: 70708BBA4183000A  AN 2)
  In Pkts Unchecked:        0
  In Pkts Delayed:          0
  In Pkts OK:               0
  In Pkts Invalid:          0
```

```
In Pkts Not Valid:         0
In Pkts Not using SA:      0
In Pkts Unused SA:         0
In Pkts Late:              0
```

Example: Show efp commands

Example: Show efp commands

Additional References for Ethernet Virtual Circuit Support for MACsec and MKA

Related Documents

Standards and RFCs

Standard/RFC	Title
Standard	*Title*

MIBs

MIB	MIBs Link
• CISCO-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

PART XI

PKI

CHAPTER 108

Cisco IOS XE PKI Overview

Cisco IOS XE public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), secure shell (SSH), and secure socket layer (SSL).

This module identifies and describes concepts that are needed to understand, plan for, and implement a PKI.

Information About Cisco IOS XE PKI

What Is Cisco IOS XE PKI

A PKI is composed of the following entities:

- Peers communicating on a secure network
- At least one certification authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

Note Public Key Infrastructure (PKI) does not support the **Inhibit Any Policy** critical extension as the internal PKI library does not recognize this extension.

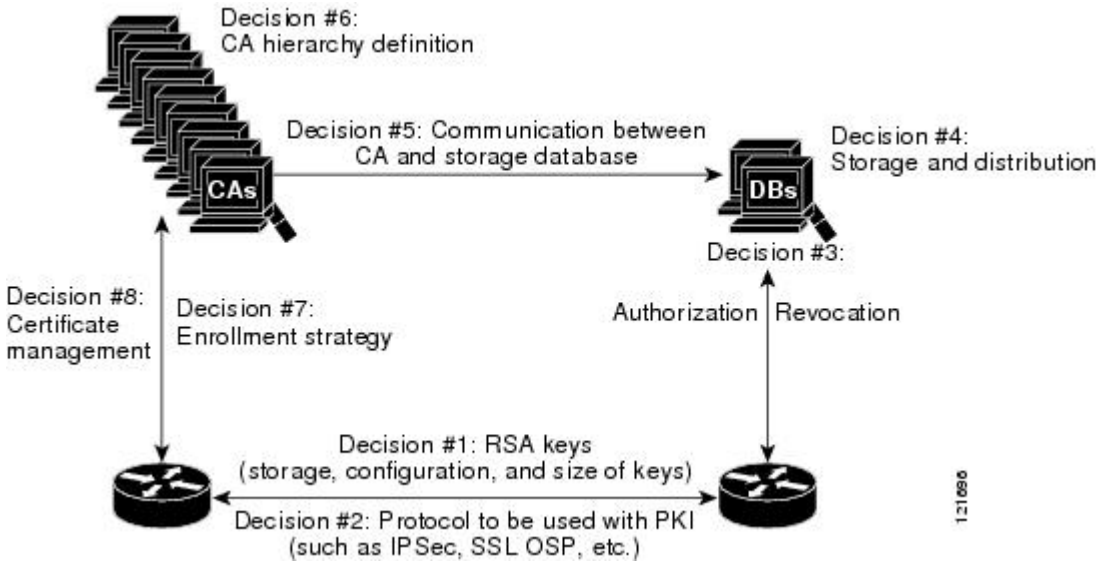
PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the

secured communicated is enrolled in the PKI in a process where the entity generates an Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Although you can plan for and set up your PKI in a number of different ways, the figure below shows the major components that make up a PKI and suggests an order in which each decision within a PKI can be made. The figure is a suggested approach; you can choose to set up your PKI from a different perspective.

Figure 34: Deciding How to Set Up Your PKI



RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

Note The default key size is 1024 bit.

What Are CAs

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

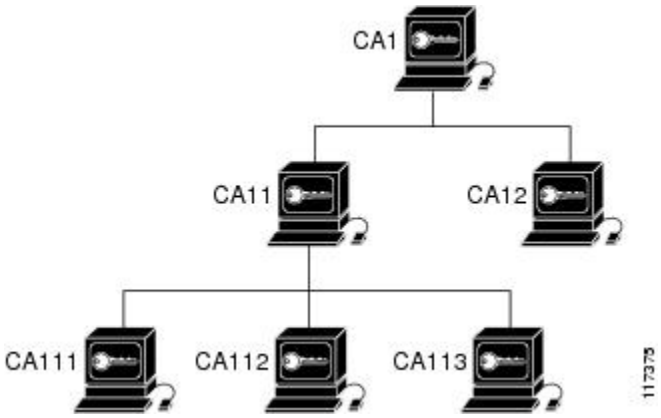
You can use a CA provided by a third-party CA vendor, or you can use an "internal" CA, which is the Cisco IOS Certificate Server.

Hierarchical PKI Multiple CAs

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. These enrollment options are how multiple tiers of CAs are configured. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

The figure below shows the enrollment relationships among CAs within a three-tiered hierarchy.

Figure 35: Three-Tiered CA Hierarchy Sample Topology



Each CA corresponds to a trustpoint. For example, CA11 and CA12 are subordinate CAs, holding CA certificates that have been issued by CA1; CA111, CA112, and CA113 are also subordinate CAs, but their CA certificates have been issued by CA11.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

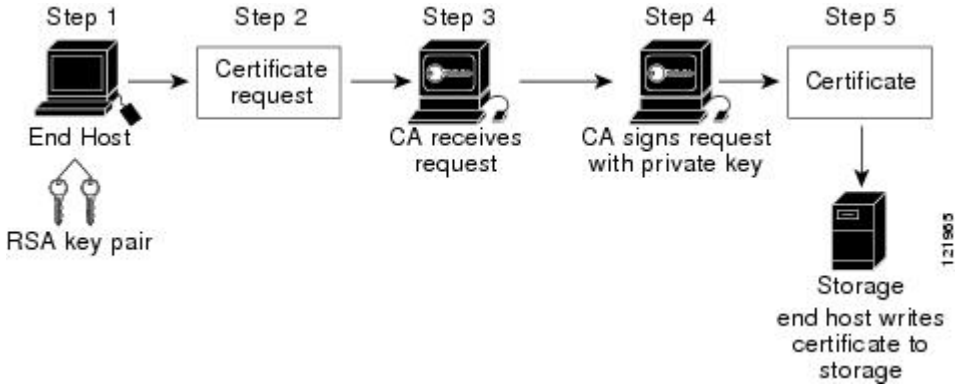
- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the CRLs.

- When online enrollment protocols are used, the root CA can be kept offline with the exception of issuing subordinate CA certificates. This scenario provides added security for the root CA.

Certificate Enrollment How It Works

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. The table below and the following steps describe the certificate enrollment process.

Figure 36: Certificate Enrollment Process



1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
   a. Manual intervention is required to approve the request.
   b. The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Note If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.

1. After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.
2. The end host writes the certificate to a storage area such as NVRAM.

Certificate Enrollment Via Secure Device Provisioning

Secure Device Provisioning (SDP) is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS XE client and a Cisco IOS certificate server.

SDP (also refer red to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a VPN. SDP involves the following three entities:

- Introducer—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
- Petitioner—A new device that is joined to the secure domain.
- Registrar—A certificate server or other server that authorizes the petitioner.

SDP is implemented over a web browser in three phases—welcome, introduction, and completion. Each phase is shown to the user via a web page.

Certificate Revocation Why It Occurs

After each participant has successfully enrolled in the PKI, the peers are ready to begin negotiations for a secure connection with each other. Thus, the peers present their certificates for validation followed by a revocation check. After the peer verifies that the other peer's certificate was issued by an authenticated CA, the CRL or Online Certificate Status Protocol (OCSP) server is checked to ensure that the certificate has not been revoked by the issuing CA. The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected.

Planning for a PKI

Planning for a PKI requires evaluating the requirements and expected use for each of the PKI components. It is recommended that you (or the network administrator) thoroughly plan the PKI before beginning any PKI configuration.

Although there are a number of approaches to consider when planning the PKI, this document begins with peer-to-peer communication. However you or the network administrator choose to plan the PKI, understand that certain decisions influence other decisions within the PKI. For example, the enrollment and deployment strategy could influence the planned CA hierarchy. Thus, it is important to understand how each component functions within the PKI and how certain component options are dependent upon decisions made earlier in the planning process.

Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module "Configuring Certificate Enrollment for a PKI."

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
PKI commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	*Configuring Certificate Enrollment for a PKI*
Certificate revocation and authorization: configuration tasks	*Configuring Revocation and Authorization of Certificates in a PKI*
Cisco IOS certificate server overview information and configuration tasks	*Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment*
Secure Device Provisioning: functionality overview and configuration tasks	*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*
Storing RSA keys and certificates on a USB eToken	*Storing PKI Credentials*

Standards and RFCs

Standard/RFC	Title
RFC 2459	*Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
RFC 2511	*Internet X.509 Certificate Request Message Format*
RFC 2527	*Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*
RFC 2528	*Internet X.509 Public Key Infrastructure*
RFC 2559	*Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2*
RFC 2560	*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

Standard/RFC	Title
RFC 2585	*Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP*
RFC 2587	*Internet X.509 Public Key Infrastructure LDAPv2 Schema*
RFC 2875	*Diffie-Hellman Proof-of-Possession Algorithms*
RFC 3029	*Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols*

MIBs

MIBs	MIBs Link
• PKI MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

CDP—certificate distribution point. Field within a digital certificate containing information that describes how to retrieve the CRL for the certificate. The most common CDPs are HTTP and LDAP URLs. A CDP may also contain other types of URLs or an LDAP directory specification. Each CDP contains one URL or directory specification.

certificates—Electronic documents that bind a user's or device's name to its public key. Certificates are commonly used to validate a digital signature.

CRL—certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

CA—certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPSec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

peer certificate--Certificate presented by a peer, which contains the peer's public key and is signed by the trustpoint CA.

PKI—public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

RA—registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

RSA keys—Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router.

CHAPTER 109

Deploying RSA Keys Within a PKI

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Configuring RSA Keys for a PKI

- Before setting up and deploying RSA keys for a PKI, you should be familiar with the module Cisco IOS PKI Overview: Understanding and Planning a PKI .

Information About RSA Keys Configuration

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs--usage keys and general-purpose keys. When you generate RSA key pairs (via the **crypto key generate rsa** command), you will be prompted to select either usage keys or general-purpose keys.

Usage RSA Keys

Usage keys consist of two RSA key pairs--one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose RSA Keys

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.

How RSA Key Pairs are Associated with a Trustpoint

A trustpoint, also known as the certificate authority (CA), manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

Caution Do not manually generate an rsa keypair under trustpoint. If we want to manually generate the keys, generate the key pairs as usage-keys and not as general-purpose keys.

Caution Certificate renewal with regenerate option does not work with key label starting from zero ('0'), (for example, '0test'). CLI allows configuring such name under trustpoint, and allows hostname starting from zero. When configuring **rsakeypair** *name* under a trustpoint, do not configure the name starting from zero. When keypair name is not configured and the default keypair is used, make sure the router hostname does not start from zero. If it does so, configure "**rsakeypair** *name* explicitly under the trustpoint with a different name.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software can maintain multiple key pairs and certificates with the same CA. As a result, the Cisco IOS software can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the **label** *key-label* option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Benefits of Exportable RSA Keys

Caution Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed. Any existing RSA keys are not exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

As of Cisco IOS Release 12.2(15)T, users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

Exportable RSA Keys in PEM-Formatted Files

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are running Cisco IOS software Release 12.3(4)T or later and who are using secure socket layer (SSL) or secure shell (SSH) applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys.

Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled “nonexportable.” To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the “exportable” keyword.

How to Set Up and Deploy RSA Keys Within a PKI

Generating an RSA Key Pair

Note We recommend that you use a new RSA keypair name for the newly configured PKI certificate. If you want to reuse an existing RSA keypair name (that is associated with an old certificate) for a new PKI certificate, do either of the following:

- Do not regenerate a new RSA keypair with an existing RSA keypair name, reuse the existing RSA keypair name. Regenerating a new RSA keypair with an existing RSA keypair name will make all the certificates associated with the existing RSA keypair invalid.
- Manually remove the old PKI certificate configurations first, before reusing the existing RSA keypair name for the new PKI certificate.

.

Perform this task to manually generate an RSA key pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]	(Optional) Generates the RSA key pair for the certificate server. • The **storage** keyword specifies the key storage location.

	Command or Action	Purpose
	Example: `Router(config)# crypto key generate rsa usage-keys modulus 2048`	• When specifying a label name by specifying the *key-label* argument, you must use the same name for the label that you plan to use for the certificate server (through the **crypto pki server** *cs-label*command). If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used. If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the **no shutdown** command, then use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate and the private key. • By default, the modulus size of a CA key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range for a modulus size of a CA key is from 360 to 4096 bits. • The **on** keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:). **Note** Keys created on a USB token must be 2048 bits or less. **Caution** Do not manually generate an rsa keypair under trustpoint. If we want to manually generate the keys, generate the key pairs as usage-keys and not as general-purpose keys.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

What to Do Next

After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).

Managing RSA Key Pairs and Trustpoint Certificates

Perform this task to configure the router to generate and store multiple RSA key pairs, associate the key pairs with a trustpoint, and get the certificates for the router from the trustpoint.

Before you begin

You must have already generated an RSA key pair as shown in the task "Generating an RSA Key Pair task."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
5. **enrollment selfsigned**
6. **subject-alt-name** *name*
7. **exit**
8. **cypto pki enroll** *name*
9. **exit**
10. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint TESTCA`	Creates a trustpoint and enters ca-trustpoint configuration mode.
Step 4	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair fancy-keys`	(Optional) The *key-label* argument specifies the name of the RSA key pair generated during enrollment (if it does not already exist or if the **auto-enroll regenerate** command is configured) to be used with the trustpoint certificate. By default, the fully qualified domain name (FQDN) key is used.

	Command or Action	Purpose
		• The keypair name cannot start from zero (‘0’). For more details, see “How RSA Key Pairs are Associated with a Trustpoint” section. • (Optional) The *key-size* argument specifies the size of the RSA key pair. The recommended key size is 2048 bits. • (Optional) The *encryption-key-size* argument specifies the size of the second key, which is used to request separate encryption, signature keys, and certificates.
Step 5	**enrollment selfsigned** **Example:** `Router(ca-trustpoint)# enrollment selfsigned`	(Optional) Specifies self-signed enrollment for a trustpoint.
Step 6	**subject-alt-name** *name* **Example:** `Router(ca-trustpoint)# subject-alt-name TESTCA`	(Optional) The *name* argument specifies the trustpoint’s name in the Subject Alternative Name (subjectAltName) field in the X.509 certificate, which is contained in the trustpoint certificate. By default, the Subject Alternative Name field is not included in the certificate. **Note** This X.509 certificate field is defined in RFC 2511. This option is used to create a self-signed trustpoint certificate for the router that contains the trustpoint name in the Subject Alternative Name (subjectAltName) field. This Subject Alternative Name can be used only when the **enrollment selfsigned** command is specified for self-signed enrollment in the trustpoint policy.

Step 7	**exit**
	Example:
	`Router`
	`(ca-trustpoint)#`
	`exit`
		Exits ca-trustpoint configuration mode.

Step 8	**cypto pki enroll** *name*
	Example:
	`Router(config)# cypto pki enroll TESTCA`
	Example:
	`% Include the router serial number in the subject name? [yes/no]: no`
	Example:
		Requests the certificates for the router from the trustpoint.
		The *name* argument specifies the trustpoint name. Once this command is entered, answer the prompts.
		Note Use the same trustpoint name entered with the **crypto pki trustpoint**command.

	Command or Action	Purpose

	`% Include an IP address in the subject name? [no]:`
	Example:
	`Generate Self Signed Router Certificate? [yes/no]: yes`
	Example:
	`Router Self Signed Certificate successfully created`	

Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 10	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Example

The following example shows how to create a self-signed trustpoint certificate for the router that contains the trustpoint name in the Subject Alternative Name (subjectAltName) field:

```
Router> enable
Router# configure terminal
Router(config)#crypto pki trustpoint TESTCA
Router(ca-trustpoint)#hash sha256
Router(ca-trustpoint)#rsakeypair testca-rsa-key 2048
Router(ca-trustpoint)#exit
Router(config)#crypto pki enroll TESTCA
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created

Router(config)#
Router(config)#exit
Router#
```

The following certificate is created:

```
Router#show crypto pki certificate verbose Router Self-Signed Certificate
   Status: Available
   Version: 3
   Certificate Serial Number (hex): 01
   Certificate Usage: General Purpose
   Issuer:
     hostname=Router.cisco.com
   Subject:
```

```
    Name: Router.cisco.com
    hostname=Router.cisco.com
  Validity Date:
    start date: 11:41:50 EST Aug 13 2012
    end   date: 19:00:00 EST Dec 31 2019
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: CA92D937 593BF19A 5B7F8466 F554D631
  Fingerprint SHA1: 57A9D411 2DDFAC81 68260F2F C6C8D7CF 4833F3E9
  X509v3 extensions:
    X509v3 Subject Key ID: 44340F76 A6B8DC37 80724650 0672875F 741D518C
    X509v3 Basic Constraints:
        CA: TRUE
    X509v3 Authority Key ID: 44340F76 A6B8DC37 80724650 0672875F 741D518C
    Authority Info Access:
  Associated Trustpoints: TESTCA

-----BEGIN CERTIFICATE-----
MIIBszCCAV2gAwIBAgIBAjANBgkqhkiG9w0BAQQFADAuMQ8wDQYDVQQDEwZURVNU
Q0ExGzAZBgkqhkiG9w0BCQIWDHIxLmNpc2NvLmNvbTAeFw0xMDAzMjIyMDI2MjBa
Fw0yMDAxMDEwMDAwMDBaMC4xDzANBgNVBAMTBlRFU1RDQTEbMBkGCSqGSIb3DQEJ
AhYMcjEuY2lzY28uY29tMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAI1xLjvrouLz
RNm8qYWI9Km9yX/wafXndY8A8o4+L8pexQhDlYyiaq7OoK6CYWH/ToyPidFW2DU0
t5WTGnIDcfsCAwEAAaNmMGQwDwYDVR0TAQH/BAUwAwEB/zARBgNVHREECjAIggZU
RVNUQ0EwHwYDVR0jBBgwFoAU+aSVh1+kyn1l+r44IFUY+Uxs1fMwHQYDVR0OBBYE
FPmklYdfpMp9Zfq+OCBVGPlMbNXzMA0GCSqGSIb3DQEBBAUAA0EAbZLnqKUaWu8T
WAIbeReTQTfJLZ8ao/U6cwXN0QKEQ37ghAdGVflFWVG6JUhv2OENNUQHXBYXNUWZ
4oBuU+U1dg==
-----END CERTIFICATE-----
```

Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you are using PKCS12 files or PEM files, exportable RSA keys allow you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

Before you begin

You must generate an RSA key pair and mark it “exportable” as specified in the “Generating an RSA Key Pair” task.

Note

- You cannot export RSA keys that existed on the router before your system was upgraded to Cisco IOS Release 12.2(15)T or later. You have to generate new RSA keys and label them as "exportable" after you upgrade the Cisco IOS software.
- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.
- If you want reexport an RSA key pair after you have already exported the key pair and imported them to a target router, you must specify the **exportable** keyword when you are importing the RSA key pair.
- The largest RSA key a router may import is 2048-bits.

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export** *trustpointname* **pkcs12** *destination-url* **password** *password-phrase*
5. **crypto pki import** *trustpointname* **pkcs12** *source-url* **password** *password-phrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint my-ca`	Creates the trustpoint name that is to be associated with the RSA key pair and enters ca-trustpoint configuration mode.
Step 2	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair my-keys`	Specifies the key pair that is to be used with the trustpoint.
Step 3	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 4	**crypto pki export** *trustpointname* **pkcs12** *destination-url* **password** *password-phrase* **Example:** `Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123`	Exports the RSA keys through the trustpoint name. • The *trustpointname* argument enters the name of the trustpoint that issues the certificate that a user is going to export. When exporting the PKCS12 file, the trustpoint name is the RSA key name.

	Command or Action	Purpose
		• The *destination-url* argument enters the file system location of the PKCS12 file to which a user wants to import the RSA key pair. • The *password -phrase* argument must be entered to encrypt the PKCS12 file for export.
Step 5	**crypto pki import** *trustpointname* **pkcs12** *source-url* **password** *password-phrase* **Example:** `Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123`	Imports the RSA keys to the target router. • The *trustpointname* argument enters the name of the trustpoint that issues the certificate that a user is going to export or import. When importing, the trustpoint becomes the RSA key name. • The *source-url* argument specifies the file system location of the PKCS12 file to which a user wants to export the RSA key pair. • The *password -phrase* must be entered to undo encryption when the RSA keys are imported.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router.

Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

Before you begin

You must generate an RSA key pair and mark it “exportable” as specified the “Generating an RSA Key Pair” task.

Note

- You cannot export and import RSA keys that were generated without an exportable flag before your system was upgraded to Cisco IOS Release 12.3(4)T or a later release. You have to generate new RSA keys after you upgrade the Cisco IOS software.
- The largest RSA key a router may import is 2048 bits.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

SUMMARY STEPS

1. **crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**]
2. **crypto pki export** *trustpoint* **pem** {**terminal** | **url** *destination-url*} {**3des** | **des**} **password** *password-phrase*
3. **crypto pki import** *trustpoint* **pem** [**check** | **exportable** | *usage-keys*] {**terminal** | **url** *source-url*} **password***password-phrase*
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**] **Example:** `Router(config)# crypto key generate rsa general-keys label mykey exportable`	Generates the RSA key pair. To use PEM files, the RSA key pair must be labeled exportable.
Step 2	**crypto pki export** *trustpoint* **pem** {**terminal** | **url** *destination-url*} {**3des** | **des**} **password** *password-phrase* **Example:** `Router(config)# crypto pki export mycs pem url nvram: 3des password mypassword123`	Exports the certificates and RSA keys that are associated with a trustpoint in a PEM-formatted file. • Enter the *trustpoint* name that is associated with the exported certificate and RSA key pair. The trustpoint name must match the name that was specified through the **crypto pki trustpoint** command • Use the **terminal** keyword to specify the certificate and RSA key pair that is displayed in PEM format on the console terminal. • Use the **url** keyword and *destination -url* argument to specify the URL of the file system where your router should export the certificates and RSA key pair. • (Optional) the **3des** keyword exports the trustpoint using the Triple Data Encryption Standard (3DES) encryption algorithm. • (Optional) the **des** keyword exports the trustpoint using the DES encryption algorithm. • Use the *password-phrase* argument to specify the encrypted password phrase that is used to encrypt the PEM file for import.

	Command or Action	Purpose
		Tip Be sure to keep the PEM file safe. For example, you may want to store it on another backup router.
Step 3	**crypto pki import** *trustpoint* **pem** [**check** | **exportable** | *usage-keys*] {**terminal** | **url** *source-url*} **password***password-phrase* **Example:** `Router(config)# crypto pki import mycs2 pem url nvram: password mypassword123`	Imports certificates and RSA keys to a trustpoint from PEM-formatted files. • Enter the *trustpoint* name that is associated with the imported certificate and RSA key pair. The trustpoint name must match the name that was specified through the **crypto pki trustpoint** command • (Optional) Use the **check** keyword to specify that an outdated certificate is not allowed. • (Optional) Use the **exportable** keyword to specify that the imported RSA key pair can be exported again to another Cisco device such as a router. • (Optional) Use the *usage-keys* argument to specify that two RSA special usage key pairs will be imported (that is, one encryption pair and one signature pair), instead of one general-purpose key pair. • Use the *source-url* argument to specify the URL of the file system where your router should import the certificates and RSA key pairs. • Use the *password-phrase* argument to specify the encrypted password phrase that is used to encrypt the PEM file for import. **Note** The password phrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser. **Note** If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:**	(Optional) Displays the RSA public keys of your router.

	Command or Action	Purpose
	`Router# show crypto key mypubkey rsa`	

Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.

Note RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also "lock" the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

Note The RSA keys must be unlocked while enrolling the CA. The keys can be locked while authenticating the router with the CA because the private key of the router is not used during authentication.

Before you begin

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in Generating an RSA Key Pair section.
- Optionally, you can authenticate and enroll each router with the CA server.

Note **Backward Compatibility Restriction**

Any image prior to Cisco IOS Release 12.3(7)T does not support encrypted keys. To prevent your router from losing all encrypted keys, ensure that only unencrypted keys are written to NVRAM before booting an image prior to Cisco IOS Release 12.3(7)T.

If you must download an image prior to Cisco IOS Release 12.3(7)T, decrypt the key and immediately save the configuration so the downloaded image does not overwrite the configuration.

Interaction with Applications

An encrypted key is not effective after the router boots up until you manually unlock the key (via the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP security (IPsec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.

>

SUMMARY STEPS

1. crypto key encrypt [write] rsa [name key-name] passphrase passphrase
2. **exit**
3. **show crypto key mypubkey rsa**
4. **crypto key lock rsa name** *key-name*] **passphrase** *passphrase*
5. **show crypto key mypubkey rsa**
6. **crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase*
7. **configure terminal**
8. **crypto key decrypt** [**write**] **rsa** [**name***key-name*] **passphrase** *passphrase*

DETAILED STEPS

	Command or Action	Purpose
Step 1	crypto key encrypt [write] rsa [name key-name] passphrase passphrase **Example:** `Router(config)# crypto key encrypt write rsa name pki.example.com passphrase password`	Encrypts the RSA keys. After this command is issued, the router can continue to use the key; the key remains unlocked. **Note** If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.
Step 2	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 3	**show crypto key mypubkey rsa** **Example:**	(Optional) Shows that the private key is encrypted (protected) and unlocked.

	Command or Action	Purpose
	`Router# show crypto key mypubkey rsa`	**Note** You can also use this command to verify that applications such as Internet Key Exchange (IKE) and SSH are properly working after the key has been encrypted.
Step 4	**crypto key lock rsa name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key lock rsa name pki.example.com passphrase password`	(Optional) Locks the encrypted private key on a running router. **Note** After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts via applications such as IKE, SSH, and SSL.
Step 6	**crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key unlock rsa name pki.example.com passphrase password`	(Optional) Unlocks the private key. **Note** After this command is issued, you can continue to establish IKE tunnels.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**crypto key decrypt** [**write**] **rsa** [**name***key-name*] **passphrase** *passphrase* **Example:** `Router(config)# crypto key decrypt write rsa name pki.example.com passphrase password`	(Optional) Deletes the encrypted key and leaves only the unencrypted key. **Note** The **write** keyword immediately saves the unencrypted key to NVRAM. If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the router is reloaded.

Removing RSA Key Pair Settings

An RSA key pair may need to be removed for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so you would have to delete the old 1024-bit keys and generate new 2048-bit keys.
- **T** he peer router's public keys can be deleted in order to help debug signature verification problems in IKEv1 and IKEv2. Keys are cached by default with the lifetime of the certificate revocation list (CRL) associated with the trustpoint.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto key zeroize rsa [key-pair-label]
4. **crypto key zeroize pubkey-chain** [*index*]
5. **exit**
6. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	crypto key zeroize rsa [key-pair-label] **Example:** `Router(config)# crypto key zeroize rsa fancy-keys`	Deletes RSA key pairs from your router. • If the *key-pair-label* argument is not specified, all RSA keys that have been generated by your router will be deleted.
Step 4	**crypto key zeroize pubkey-chain** [*index*] **Example:** `Router(config)# crypto key zeroize pubkey-chain`	Deletes the remote peer's public key from the cache. (Optional) Use the *index* argument to delete a particular public key index entry. If no index entry is specified, then all the entries are deleted. The acceptable range of index entries is from 1 to 65535.
Step 5	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 6	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Configuration Examples for RSA Key Pair Deployment

Generating and Specifying RSA Keys Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair "exampleCAkeys":

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

Exporting and Importing RSA Keys Examples

Exporting and Importing RSA Keys in PKCS12 Files Example

In the following example, an RSA key pair "mynewkp" is generated on Router A, and a trustpoint name "mynewtp" is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so that it can be imported on Router B. By importing the trustpoint "mynewtp" to Router B, the user has imported the RSA key pair "mynewkp" to Router B.

Router A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair mykeys
 exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

Router B

```
crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.
!
July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

Exporting and Importing and RSA Keys in PEM Files Example

The following example shows the generation, exportation, and importation fo the RSA key pair "mytp", and verifies its status:

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mytp exportable

The name for the keys will be: mytp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto pki export mytp pem url nvram:mytp 3des password mypassword123

% Key name:mytp
Usage:General Purpose Key
Exporting public key...
Destination filename [mytp.pub]?
Writing file to nvram:mytp.pub
Exporting private key...
Destination filename [mytp.prv]?
Writing file to nvram:mytp.prv
!
! Import the key as a different name.
!
Router(config)# crypto pki import mytp2 pem url nvram:mytp2 password mypassword123

% Importing public key or certificate PEM file...
Source filename [mytp2.pub]?
Reading file from nvram:mytp2.pub
% Importing private key PEM file...
Source filename [mytp2.prv]?
Reading file from nvram:mytp2.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2011
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
```

```
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2011
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
```

Exporting Router RSA Key Pairs and Certificates from PEM Files Example

The following example shows how to generate and export the RSA key pair "aaa" and certificates of the router in PEM files that are associated with the trustpoint "mycs." This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

```
Router(config)# crypto key generate rsa general-keys label aaa exportable

The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs

Router(ca-trustpoint)# enrollment url http://mycs

Router(ca-trustpoint)#
rsakeypair aaa

Router(ca-trustpoint)# exit

Router(config)# crypto pki authenticate mycs

Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs

%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:
Re-enter password:
% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be:host.example.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.
```

```
Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157
00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority
Router(config)# crypto ca export aaa pem terminal 3des password

% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----
% Key name:aaa
Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,ED6B210B626BC81A
Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbAAUpGk7VnPCT87
<snip>
kLCOtxzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----
% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfigAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
<snip>
6xlBaIsuMxnHmr89KkKkYlU6
-----END CERTIFICATE-----
```

Importing Router RSA Key Pairs and Certificate from PEM Files Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint “ggg” from PEM files via TFTP:

```
Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password

% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via Ethernet0):!
[OK - 1082 bytes]
% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via Ethernet0):!
[OK - 573 bytes]
% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via Ethernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#
```

Encrypting and Locking Private Keys on a Router Examples

Configuring and Verifying an Encrypted Key Example

The following example shows how to encrypt the RSA key “pki-123.example.com.” Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki-123.example.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa
```

% Key pair was generated at:00:15:32 GMT Jun 25 2003

Key name:pki-123.example.com

Usage:General Purpose Key

*** The key is protected and UNLOCKED. ***

Key is not exportable.

Key Data:

305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C

CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFEE340 0EC1E62B D40E7DCC

23C4D09E

03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001

% Key pair was generated at:00:15:33 GMT Jun 25 2003

Key name:pki-123.example.com.server

Usage:Encryption Key

Key is exportable.

Key Data:

307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00D3491E 2A21D383

854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FFF5757

3A41CE04 FDCB40A4 B9C68B4F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4

DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001

```
Router#
```

Configuring and Verifying a Locked Key Example

The following example shows how to lock the key "pki-123.example.com." Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki-123.example.com passphrase password
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki-123.example.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

Additional References

Related Documents

Related Topic	Document Title
Overview of PKI, including RSA keys, certificate enrollment, and CAs	Cisco IOS PKI Overview: Understanding and Planning a PKI
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Recommended cryptographic algorithms	*Next Generation Encryption*

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2409	*The Internet Key Exchange (IKE)*
RFC 2511	Internet X.509 Certificate Request Message Format

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 146: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 110

Configuring Authorization and Revocation of Certificates in a PKI

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI). It includes information on high-availability support for the certificate server.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the certificate authority (CA).
- Enrolled peer devices with the CA.
- Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

High Availability

For high availability, IPsec-secured Stream Control Transmission Protocol (SCTP) must be configured on both the active and the standby routers. For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

Restrictions for Authorization and Revocation of Certificates

- PKI High Availability (HA) support of intra-chassis stateful switchover (SSO) redundancy is currently not supported on all switches running the Cisco IOS Release 12.2 S software. See Cisco bug CSCtb59872 for more information.
- Depending on your Cisco IOS release, Lightweight Directory Access Protocol (LDAP) is supported.

Information About Authorization and Revocation of Certificates

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server.

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of "all" from the AAA server provides authorization. Likewise, a label of "none" from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but "none" is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be "ipsec," "ssl," or "osp." (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

Note Currently, no application component supports specification of the application label.

- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

RADIUS or TACACS+ Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to "cisco," which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute "cert-application=all" is added under the PKI service for the particular user or usergroup to authorize the specific username.

Attribute-Value Pairs for PKI and AAA Server Integration

The table below lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

Note Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The **all** parameter (within the **authorization username** command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 147: AV Pairs That Must Match

AV Pair	Value
cisco-avpair=pki:cert-application=all	Valid values are "all" and "none."

AV Pair	Value
cisco-avpair=pki:cert-trustpoint=msca	The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label. **Note** The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-serial=16318DB7000100001671	The value is a certificate serial number. **Note** The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year. **Note** Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).

CRLs or OCSP Server Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms--certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the PKI and AAA Server Integration for Certificate Status section.

The following sections explain how each revocation mechanism works:

What Is a CRL

A certificate revocation list (CRL) is a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL is downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration applies to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router does not know that the certificate has been revoked. The certificate passes the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device uses the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified through the **cdp-url**command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes.
- The CRL lifetime determines the length of time between CA-issued updates to the CRL. The default CRL lifetime value, which is 168 hours [1 week], can be changed through the **lifetime crl** command.
- The method of the CDP determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP. HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
- The location of the CDP determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Note If Public Key Infrastructure (PKI) with Certificate Revocation Lists (CRLs) are used, and if the size of the PKI CRL file exceeds 200 KB (approximately) and above, a CPU HOG may be generated.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected. To prevent a possible certificate rejection and if there are multiple CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.

Note Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.

Tip Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:

- OCSP can provide real-time certificate status checking.
- OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
- OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
- OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:

- Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.
- There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS memory and may reduce resources available to other processes.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

Note As of Cisco IOS Release 12.4(9)T or later, an administrator may configure CRL caching, either by disabling CRL caching completely or setting a maximum lifetime for a cached CRL per trustpoint.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value--equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the **match certificate**command with the **skip revocation-check** keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the **match certificate**command with the **skip revocation-check** keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

Ignoring Expired Certificates

To configure your router to ignore expired certificates, enter the **match certificate** command with the **allow expired-certificate** keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to “allow” the expired certificate until the peer can obtain a new certificate.

- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

Note If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be "brought up" because the certificate of the hub is not yet valid.

- "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **match certificate** command with the **skip authorization-check** keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **match certificate** command with the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **match certificate**command and the **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

Note If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be "brought up" because the certificate of the AAA server is not yet valid.

PKI Certificate Chain Validation

A certificate chain establishes a sequence of trusted certificates --from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS Release 12.4(6)T and later releases, an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

Reauthentication of Trusted Certificates

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that

the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

Extending the Trusted Certificate Chain

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer's certificate chain and the router's trusted certificates are validated to a specified point.

Completing Gaps in a Certificate Chain

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.

Note If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

Note It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

How to Configure Authorization and Revocation of Certificates for Your PKI

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

Note The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.
- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least significant RDN first) is used.

or

radius-server host *hostname* [**key** *string*]

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization network** *listname* [*method*]
5. **crypto pki trustpoint** *name*
6. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
7. revocation-check method
8. **exit**
9. **authorization username subjectname** *subjectname*
10. **authorization list** *listname*
11. **tacacs-server host** hostname [**key** string]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization network** *listname* [*method*] **Example:** `Router (config)# aaa authorization network maxaaa group tacacs+`	Sets the parameters that restrict user access to a network. • *method* --Can be **group radius**, **group tacacs+**, or **group group-name**.
Step 5	**crypto pki trustpoint** *name* **Example:** `Route (config)# crypto pki trustpoint msca`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 6	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
	Example:
	`Router (ca-trustpoint)# enrollment url http://caserver.myexample.com`
	- or-
	`Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80`
		Specifies the following enrollment parameters of the CA:
		• (Optional) The **mode** keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.
		• (Optional) The **retry period** keyword and *minutes* argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1.
		• (Optional) The **retry count** keyword and *number* argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10.
		• The *url* argument is the URL of the CA to which your router should send certificate requests.

	Command or Action	Purpose
		Note With the introduction of Cisco IOS Release 15.2(1)T, an IPv6 address can be added to the **http:** enrolment method. For example: http://[ipv6-address]:80. The IPv6 address must be enclosed in brackets in the URL. See the Command Reference document for more information on the other enrollment methods that can be used. • (Optional) The **pem** keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 7	revocation-check method **Example:** `Router (ca-trustpoint)# revocation-check crl`	(Optional) Checks the revocation status of a certificate.
Step 8	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	**authorization username subjectname** *subjectname* **Example:** `Router (config)# authorization username subjectname serialnumber`	Sets parameters for the different certificate fields that are used to build the AAA username. The *subjectname* argument can be any of the following: • **all** --Entire distinguished name (subject name) of the certificate. • **commonname** --Certification common name. • **country** --Certificate country. • **email** --Certificate e-mail. • **ipaddress** --Certificate IP address. • **locality** --Certificate locality. • **organization** --Certificate organization. • **organizationalunit** --Certificate organizational unit. • **postalcode** --Certificate postal code. • **serialnumber** --Certificate serial number. • **state** --Certificate state field. • **streetaddress** --Certificate street address. • **title** --Certificate title.

	Command or Action	Purpose
		• **unstructuredname** --Certificate unstructured name.
Step 10	**authorization list** *listname* **Example:** `Route (config)# authorization list maxaaa`	Specifies the AAA authorization list.

Step 11	**tacacs-server host** hostname [**key** string]
	Example:
	`Router(config)# tacacs-server host 192.0.2.2 key a_secret_key`
	Example:
	`radius-server host hostname [key string]`
	Example:
	`Router(config)# radius-server host 192.0.2.1 key another_secret_key`
		Specifies a TACACS+ host.
		or
		Specifies a RADIUS host.

Troubleshooting Tips

To display debug messages for the trace of interaction (message type) between the CA and the router, use the **debug crypto pki transactions**command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

Successful Exchange

```
Router# debug crypto pki transactions
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
```

Each line that shows “CRYPTO_PKI_AAA” indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

```
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aaalist,
PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

Failed Exchange

```
Router# debug crypto pki transactions
Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"= "CA1")
```

```
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "233D")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism--CRLs or OCSP--that is used to check the status of certificates in a PKI.

The revocation-check Command

Use the **revocation-check** command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer's certificate--unless you include the **none** keyword in your configuration. If the **none** keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Note When the revocation check is changed to 'none' under a trustpoint, the CRL cache associated with the CA certificate of the trustpoint will be cleared.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.

Before you begin

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.

Note

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.
- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki trustpoint *name*
4. **ocsp url** *url*
5. **revocation-check** *method1* [*method2 method3*]]
6. **ocsp disable-nonce**
7. **exit**
8. **exit**
9. **show crypto pki certificates**
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	crypto pki trustpoint *name* **Example:** `Router(config)# crypto pki trustpoint hazel`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 4	**ocsp url** *url*
	Example:
	`Router(ca-trustpoint)# ocsp url http://ocsp-server`
	- or -
	`Router(ca-trustpoint)# ocsp url http://10.10.10.1:80`
		The *url* argument specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL overrides the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. All certificates associated with a configured trustpoint are checked by the OCSP server. The URL can be a hostname, IPv4 address, or an IPv6 address.

	Command or Action	Purpose
	- or - `Router(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80`	
Step 5	**revocation-check** *method1* [*method2 method3*]] **Example:** `Router(ca-trustpoint)# revocation-check ocsp none`	Checks the revocation status of a certificate. • **crl** --Certificate checking is performed by a CRL. This is the default option. • **none** --Certificate checking is ignored. • **ocsp** --Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.
Step 6	**ocsp disable-nonce** **Example:** `Router(ca-trustpoint)# ocsp disable-nonce`	(Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.
Step 7	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 9	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints`	Displays information about the trustpoint configured in router.

Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate command** and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.

Note Certificate maps are checked even if the peer's public key is cached. For example, when the public key is cached by the peer, and a certificate map is added to the trustpoint to ban a certificate, the certificate map is effective. This prevents a client with the banned certificate, which was once connected in the past, from reconnecting.

Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Manually Overriding the OCSP Server Setting in a Certificate

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the **ocsp url** command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the **match certificate override ocsp** command. The **match certificate override ocsp**command overrides the client certificate AIA field or the **ocsp url**command setting if a client certificate is successfully matched to a certificate map during the revocation check.

Note Only one OCSP server can be specified per client certificate.

Configuring CRL Cache Control

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the **crl cache delete-after** command or disable CRL caching by issuing the **crl cache none** command. Only the **crl-cache**

delete-aftercommand or the **crl-cache none** command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the **crl-cache none** command nor the **crl-cache delete-after** command affects the currently cached CRL. If you configure the **crl-cache none** command, all CRLs downloaded after this command is issued will not be cached. If you configure the **crl-cache delete-after** command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful is when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

Configuring Certificate Serial Number Session Control

A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the **serial-number** field or an AAA attribute, with the **cert-serial-not** command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.

Before you begin

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled or the **serial-number** command is issued.
- The PKI and AAA server integration must be successfully completed to use AAA attributes as described in "PKI and AAA Server Integration for Certificate Status."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki certificate map label sequence-number
4. *field-name match-criteria match-value*
5. **exit**
6. **crypto pki trustpoint** *name*
7. Do one of the following:
   - **crl-cache none**
   - **crl-cache delete-after** *time*
8. **match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check**
9. **match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*
10. **match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url*
11. **exit**
12. **aaa new-model**

13. **aaa attribute list** *list-name*
14. **attribute type** {*name*} {*value*}
15. **exit**
16. **exit**
17. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	crypto pki certificate map label sequence-number **Example:** `Router(config)# crypto pki certificate map Group 10`	Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.
Step 4	*field-name match-criteria match-value* **Example:** `Router(ca-certificate-map)# subject-name co MyExample`	Specifies one or more certificate fields together with their matching criteria and the value to match. The *field-name* is one of the following case-insensitive name strings or a date: • **alt-subject-name** • **expires-on** • **issuer-name** • **name** • **serial-number** • **subject-name** • **unstructured-subject-name** • **valid-start** **Note** Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss. The *match-criteria* is one of the following logical operators:

	Command or Action	Purpose
		• **co** --contains (valid only for name fields and serial number field) • **eq** --equal (valid for name, serial number, and date fields) • **ge** --greater than or equal (valid only for date fields) • **lt** --less than (valid only for date fields) • **nc** --does not contain (valid only for name fields and serial number field) • **ne** --not equal (valid for name, serial number, and date fields) The *match-value*is the name or date to test with the logical operator assigned by match-criteria. **Note** Use this command only when setting up a certificate-based ACL--not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.
Step 5	**exit** **Example:** `Router(ca-certificate-map)# exit`	Returns to global configuration mode.
Step 6	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Access2`	Declares the trustpoint, given name and enters ca-trustpoint configuration mode.

Step 7	Do one of the following:
	• **crl-cache none**
	• **crl-cache delete-after** *time*
	Example:
	`Router(ca-trustpoint)# crl-cache none`
	Example:
	`Router(ca-trustpoint)# crl-cache delete-after 20`
		(Optional) Disables CRL caching completely for all CRLs associated with the trustpoint.
		The **crl-cache none** command does not affect any currently cached CRLs. All CRLs downloaded after this command is configured will not be cached.
		(Optional) Specifies the maximum time CRLs will remain in the cache for all CRLs associated with the trustpoint.
		• *time* --The amount of time in minutes before the CRL is deleted.
		The **crl-cache delete-after** command does not affect any currently cached CRLs. The configured lifetime will only affect CRLs downloaded after this command is configured.

	Command or Action	Purpose
Step 8	**match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check** **Example:** `Router(ca-trustpoint)# match certificate Group skip revocation-check`	(Optional) Associates the certificate-based ACL (that was defined via the **crypto pki certificate map** command) to a trustpoint. • *certificate-map-label* --Must match the *label* argument specified via the **crypto pki certificate map** command. • **allow expired-certificate** --Ignores expired certificates. • **skip revocation-check** --Allows a trustpoint to enforce CRLs except for specific certificates. • **skip authorization-check** --Skips the AAA check of a certificate when PKI integration with an AAA server is configured.
Step 9	**match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string* **Example:** `Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com`	(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label* --A user-specified label that must match the *label* argument specified in a previously defined **crypto pki certificate map** command. • **url** --Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory** --Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string* --The URL or directory specification. **Note** Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.
Step 10	**match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url* **Example:** `Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url http://192.0.2.2`	(Optional) Specifies an OCSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OCSP servers and client certificate settings including alternative PKI hierarchies. • *certificate-map-label* --The name of an existing certificate map. • **trustpoint** --The trustpoint to be used when validating the OCSP server certificate.

	Command or Action	Purpose
		• *sequence-number* --The order the **match certificate override ocsp** command statements apply to the certificate being verified. Matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, it overwrites the previous OCSP server override setting. • **url** --The URL of the OCSP server. When the certificate matches a configured certificate map, the AIA field of the client certificate and any previously issued **ocsp url** command settings are overwritten with the specified OCSP server. If no map-based match occurs, one of the following two cases will continue to apply to the client certificate. • If OCSP is specified as the revocation method, the AIA field value will continue to apply to the client certificate. • If the **ocsp url** configuration exists, the **ocsp url** configuration settings will continue to apply to the client certificates.
Step 11	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 12	**aaa new-model** **Example:** `Router(config)# aaa new-model`	(Optional) Enables the AAA access control model.
Step 13	**aaa attribute list** *list-name* **Example:** `Router(config)# aaa attribute list crl`	(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode.
Step 14	**attribute type** {*name*} {*value*} **Example:** `Router(config-attr-list)# attribute type cert-serial-not 6C4A`	(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router. To configure certificate serial number session control, an administrator may specify a specific certificate in the *value* field to be accepted or rejected based on its serial number where *name* is set to **cert-serial-not**. If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected.

	Command or Action	Purpose
		For a full list of available AAA attribute types, execute the **show aaa attributes** command.

Step 15	**exit**
	Example:
	`Router(ca-trustpoint)# exit`
	Example:
	`Router(config-attr-list)# exit`
		Returns to global configuration mode.

Step 16	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 17	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.

Example

The following is a sample certificate. The OCSP-related extensions are shown using exclamation points.

```
Certificate:
        Data:
            Version: v3
            Serial Number:0x14
            Signature Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Issuer:CN=CA server,OU=PKI,O=Cisco Systems
            Validity:
                Not Before:Thursday, August 8, 2002 4:38:05 PM PST
                Not After:Tuesday, August 7, 2003 4:38:05 PM PST
            Subject:CN=OCSP server,OU=PKI,O=Cisco Systems
            Subject Public Key Info:
                Algorithm:RSA - 1.2.840.113549.1.1.1
                Public Key:
                    Exponent:65537
                    Public Key Modulus:(2048 bits) :
                        <snip>
            Extensions:
                Identifier:Subject Key Identifier - 2.5.29.14
                    Critical:no
                    Key Identifier:
                        <snip>
                Identifier:Authority Key Identifier - 2.5.29.35
                    Critical:no
                    Key Identifier:
                        <snip>
!                Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
                     Critical:no
```

```
                Identifier:Extended Key Usage:- 2.5.29.37
                     Critical:no
                     Extended Key Usage:
                     OCSPSigning
!
                Identifier:CRL Distribution Points - 2.5.29.31
                    Critical:no
                    Number of Points:1
                    Point 0
                        Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
        Signature:
            Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Signature:
            <snip>
```

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocsp** command to the beginning of an existing sequence:

```
match certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
.
.
.
        match certificate map3 override ocsp 5 url http://192.0.2.3/
        match certificate map1 override ocsp 10 url http://192.0.2.1/
        match certificate map2 override ocsp 15 url http://192.0.2.2/
```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocsp** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.0.2.3/
        match certificate map1 override ocsp trustpoint tp1 10 url http://192.0.2.1/
        match certificate map4 override ocsp trustpoint tp4 10 url
http://192.0.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.0.2.2/
```

Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuring Certificate Chain Validation

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

Before you begin

- The device must be enrolled in your PKI hierarchy.

- The appropriate key pair must be associated with the certificate.

Note

- A trustpoint associated with the root CA cannot be configured to be validated to the next level.

The **chain-validation** command is configured with the **continue** keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default **chain-validation**command setting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki trustpoint *name*
4. **chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	crypto pki trustpoint *name* **Example:** `Router(config)# crypto pki trustpoint ca-sub1`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**chain-validation** [{**stop**	**continue**} [*parent-trustpoint*]] **Example:** `Router(ca-trustpoint)# chain-validation continue ca-sub1`	Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates. • Use the **stop**keyword to specify that the certificate is already trusted. This is the default setting. • Use the **continue** keyword to specify that the that the subordinate CA certificate associated with the trustpoint must be validated. • The *parent-trustpoint* argument specifies the name of the parent trustpoint the certificate must be validated against.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode

Configuring CRL Autodownload

Perform this step to configure the certificate revocation list (CRL) autodownload.

Improper configuration of this feature can enable excessive CRL downloads for CRLs already cached by the device thereby halting validations because the CRL download and CRL validation cannot be executed in parallel. If a CRL is already downloaded, the downloaded CRL can be used for certificate validation without downloading additional CRLs.

If you configure the **crl-cache none** command, you cannot auto download a CRL for a trustpoint. To download the CRL, execute the **no crl cache none** command to remove the CRL cache from trustpoint. Similarly, when a CRL download is configured, you cannot enable the **crl-cache none** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki crl download url** *url* [**source-interface** *interface-name* | **vrf** *vrf-name*]
4. **crypto pki crl download trustpoint** *trustpoint-label*
5. **crypto pki crl download schedule time** *day hh*:*ss*
6. **crypto pki crl download schedule prepublish** *minutes*
7. **crypto pki crl download schedule retries** *number* **crypto pki crl download schedule retries interval** *minutes*
8. **end**
9. **crypto pki crl refresh-cache**
10. **show crypto pki crl download**
11. **show crypto pki timers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki crl download url** *url* [**source-interface** *interface-name* | **vrf** *vrf-name*] **Example:** `Device(config)# crypto pki crl download url www.abc.com source-interface GigabitEthernet 1`	Specifies that the CRL auto download must fetch the CRL through the source interface or the VRF or both.
Step 4	**crypto pki crl download trustpoint** *trustpoint-label* **Example:** `Device(config)# crypto pki crl download trustpoint trp1`	Specifies that the CRL auto download must fetch the CRL distribution point (CDP) from the device certificate associated with that trustpoint.
Step 5	**crypto pki crl download schedule time** *day hh*:*ss* **Example:** `Device(config)# crypto pki crl download schedule time Monday 00:00`	Specifies the day and time when the CRL auto download must be triggered. • *time*—Indicates the exact time of the day to download the CRL, if no CRL is found. Must be specified in hour and minute format (*mm*:*ss*).
Step 6	**crypto pki crl download schedule prepublish** *minutes* **Example:** `Device(config)# crypto pki crl download schedule prepublish 720`	Time interval, in minutes, to download the CRL before the CRL expires. the default value is 0.
Step 7	**crypto pki crl download schedule retries** *number* **crypto pki crl download schedule retries interval** *minutes* **Example:** `Device(config)# crypto pki crl download schedule retries 15 interval 15 crypto pki crl download schedule retries 15 interval 15`	Specifies the time interval, in minutes, for a device to retry downloading a CRL from a CDP location if previous download attempts fail. The default number of retries is 5. • **interval** *minutes*—Time interval between retry attempts, in minutes. The default retry interval is 30 minutes.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 9	**crypto pki crl refresh-cache** **Example:** `Device# crypto pki crl refresh-cache`	Refreshes the CRL entries in the cache.
Step 10	**show crypto pki crl download** **Example:** `Device# show crypto pki crl download`	Displays auto download configurations.
Step 11	**show crypto pki timers** **Example:** `Device(config)# show crypto pki timers`	Displays information about the timers set for Cisco IOS for public key infrastructure.

Example

The following is a sample output from the **show crypto pki crl download** command.

```
Device# show crypto pki crl download

CRL Issuer Name:
    cn=ios
    LastUpdate: 10:38:23 IST Sep 18 2013
    NextUpdate: 16:38:23 IST Sep 18 2013

    Valid after expiry till: 16:58:23 IST Sep 18 2013

    CRL Downloaded at 12:38:23 IST Sep 18 2013

    Retrieved from CRL Distribution Point:
      ** CDP Not Published - Retrieved via SCEP

 CRL DER is 213 bytes
 CRL is stored in parsed CRL cache

CRL prepublish timer interval: 10

Parsed CRL cache current size is 213 bytes
Parsed CRL cache maximum size is 65536 bytes
```

- The field Valid after expiry till: indicates the duration for which the CRL is valid after expiry when crl cache extend is configured.
- The field CRL Downloaded at denotes the time when the CRL is downloaded.

The following is a sample output from the **show crypto pki timer** command.

```
Device# show crypto pki timers

PKI Timers
  |       13:42.564
  |       13:42.564  SESSION CLEANUP
  |       11:44.111
  |       11:44.111  CRL UPDATE cn=IOS-CA
  |       21:44.111  CRL EXPIRE cn=IOS-CA
  |    7:59:56.917  STATIC CRL DOWNLOAD
CS Timers
  |        1:44.071
  |        1:44.071  CS DB CLEANUP
  |       11:43.999  CS SHADOW CERT GENERATION
  |       21:43.883  CS CERT EXPIRE
```

The field CRL UPDATE denotes the updated timer based on the prepublish time.

Configuration Examples for Setting Up Authorization and Revocation of Certificates

Configuring and Verifying PKI AAA Authorization Examples

This section provides configuration examples of PKI AAA authorizations:

Router Configuration Example

The following **show running-config**command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config
Building configuration...
!
version 12.3
!
hostname router7200router7200
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
 enrollment url http://192.0.2.33:80
 serial-number
 crl optional
 rsakeypair STOREVPN 2048
 auto-enroll
 authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
 certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
  31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
  55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
  312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
  30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
  7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
  FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
  16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
  030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
  341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
  12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
  08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
  15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
  EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
  31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
  55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
  589223AB 99A7DC14 04F74EF2 AAEEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
  54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
  E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
```

```
  22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
  30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
  F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
  BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
  0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
  12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
  3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
 encr aes
 group 14
!
!
crypto ipsec transform-set ISC_TS_1 esp-aes esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
 set security-association lifetime kilobytes 530000000
 set security-association lifetime seconds 14400
 set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
 description MGRE Interface provisioned by ISC
 bandwidth 10000
 ip address 192.0.2.172 255.255.255.0
 no ip redirects
 ip mtu 1408
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 500
 ip nhrp server-only
 no ip split-horizon eigrp 101
 tunnel source FastEthernet2/1
 tunnel mode gre multipoint
 tunnel key 101
 tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
 ip address 192.0.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 ip address 192.0.2.2 255.255.255.0
 duplex auto
 speed auto
!
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end
```

Debug of a Successful PKI AAA Authorization Example

The following **show debugging** command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
Crypto PKI Trans debugging is on
Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD5.example.com)
May 28 19:36:12.813: TPLUS: Using server 192.0.2.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is
 up: new adjacency
Router#
Router# show crypto isakmp sa
dst             src             state          conn-id slot
192.0.2.22     192.0.2.102   QM_IDLE             84    0
```

Debugs of a Failed PKI AAA Authorization Example

The following **show debugging** command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
```

```
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing
May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD5.example.com)
May 28 19:48:31.533: TPLUS: Using server 192.0.2.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD5.example.com)
May 28 19:48:41.505: TPLUS: Using server 198.168.244.55
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
Router#
Router# show crypto iskmp sa
```

```
dst             src             state          conn-id slot
192.0.2.2        192.0.2.102   MM_KEY_EXCH         95    0
```

Configuring a Revocation Mechanism Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:

Configuring an OCSP Server Example

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

Specifying a CRL and Then an OCSP Server Example

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specifying an OCSP Server Example

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Disabling Nonces in Communications with the OCSP Server Example

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
Router(ca-trustpoint)# ocsp disable-nonce
```

Configuring a Hub Router at a Central Site for Certificate Revocation Checks Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration--only the PKI-related configuration is shown.

Home Office Hub Configuration

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Central Site Hub Router

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Central VPN Gateway
    cn=Central VPN Gateway
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Sep 26 2004
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: VPN-GW
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: VPN-GW
```

Trustpoint on the Branch Office Router

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
```

```
ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
```

A certificate map is entered on the branch office router.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

The output from the **show certificate** command on the central site hub router shows that the certificate was issued by the following:

```
cn=Central Certificate Authority
o=Home Office Inc
```

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be shown on one line with the line above it.
```

The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with “Name:” is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
 match certificate central-site skip revocation-check
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
```

```
 subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the **match certificate**command and **central-site skip authorization-check (argument and keyword)**, the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate**command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Trustpoint on the Branch 1 Site Router

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Oct 3 2003
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
```

```
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office
```

A certificate map is entered on the central site router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Brahcn 1 Site,o=home office inc
```

The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) #exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term
!many lines left out
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

The **match certificate**command and **branch1 allow expired-certificate** (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Configuring Certificate Authorization and Revocation Settings Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache none
```

The current CRL is still cached immediately after executing the example configuration shown above:

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The **crl-cache none**command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the **show crypto pki crls** command. No output will be shown because there are no CRLs cached.

The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache delete-after 2
```

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
When the current CRL expires, a new CRL is downloaded to the router at the next update and
 the crl-cache delete-after
command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after
a maximum lifetime of 2 minutes.
You can verify that the CRL will be cached for 2 minutes by executing the show crypto pki
crls
 command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.
```

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
```

```
    LastUpdate: 22:57:42 GMT Nov 26 2005

    NextUpdate: 22:59:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
```

ldap://ldap.example.com/CN=name Cert Manager,O=example.com

Configuring Certificate Serial Number Session Control

The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 chain-validation stop
 crl query ldap://ldap_server
 revocation-check crl
 match certificate crl
!
crypto pki certificate map crl 10
 serial-number co 279d
```

Note If the *match-criteria* value is set to **eq** (equal) instead of **co** (contains), the serial number must match the certificate map serial number exactly, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number "4ACA."

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 aaa new-model
!
aaa attribute list crl
attribute-type aaa-cert-serial-not 4ACA
```

The server log shows that the certificate with the serial number "4ACA" was rejected. The certificate rejection is shown using exclamation points.

```
.
.
.
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: validation path has 1 certs
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: Certificate validated without revocation check
Dec 3 04:24:39.135: CRYPTO_PKI: Selected AAA username: 'PKIAAA'
Dec 3 04:24:39.135: CRYPTO_PKI: Anticipate checking AAA list:'CRL'
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: checking AAA authorization (CRL, PKIAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x4)
```

```
Dec 3 04:24:39.135: AAA/BIND(00000021): Bind i/f
Dec 3 04:24:39.135: AAA/AUTHOR (0x21): Pick method list 'CRL'
.
.
.
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: cert-serial doesn't match ("4ACA" != "4ACA")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: post-authorization chain validation status (0x7)
!
Dec 3 04:24:39.175: CRYPTO_PKI: AAA authorization for list 'CRL', and user 'PKIAAA' failed.
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain
validation result was: CRYPTO_PKI_CERT_NOT_AUTHORIZED
!
Dec 3 04:24:39.175: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.43 is bad:
 certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer
at 192.0.2.43
.
.
.
```

Configuring Certificate Chain Validation Examples

This section contains the following configuration examples that can be used to specify the level of certificate chain processing for your device certificates:

Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated--the peer, SubCA11, SubCA1, and RootCA certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated--the peer and SubCA1 certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
```

```
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated--the peer, SubCA11, and SubCA1 certificates.

If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA11
```

Additional References

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Overview of PKI, including RSA keys, certificate enrollment, and CAs	“Cisco IOS PKI Overview: Understanding and Planning a PKI” module
RSA key generation and deployment	“Deploying RSA Keys Within a PKI” module
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	“Configuring Certificate Enrollment for a PKI” module
Cisco IOS certificate server overview information and configuration tasks	“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment ” module
Recommended cryptographic algorithms	*Next Generation Encryption*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 148: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 111

Configuring Certificate Enrollment for a PKI

This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer. Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host that requests the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for PKI Certificate Enrollment

Before configuring peers for certificate enrollment, you should have the following items:

- A generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.
- An authenticated CA.
- Familiarity with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."
- Enable NTP on the device so that the PKI services such as auto enrollment and certificate rollover may function correctly.

Note As of Cisco IOS Release 12.3(7)T, all commands that begin with "**crypto ca**" have been changed to begin with "**crypto pki**." Although the router will still accept **crypto ca** commands, all output will be be displayed **crypto pki**.

Information About Certificate Enrollment for a PKI

What Are CAs

A CA is an entity that issues digital certificates that other parties can use. It is an example of a trusted third party. CAs are characteristic of many PKI schemes.

A CA manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use the Cisco IOS certificate server or a CA provided by a third-party CA vendor.

Framework for Multiple CAs

A PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the certificate revocation lists (CRLs).
- When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

Authentication of the CA

The certificate of the CA must be authenticated before the device will be issued its own certificate and before certificate enrollment can occur. Authentication of the CA typically occurs only when you initially configure PKI support at your router. To authenticate the CA, issue the **crypto pki authenticate** command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

Note PKI does not support certificate with lifetime validity greater than the year 2099. So, It is recommended to choose a life time validity fewer than the value 2099.

Authentication via the fingerprint Command

Cisco IOS Release 12.3(12) and later releases allow you to issue the **fingerprint** command t o preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

If a fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

Note If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.

Supported Certificate Enrollment Methods

Cisco IOS software supports the following methods to obtain a certificate from a CA:

- Simple Certificate Enrollment Protocol (SCEP)--A Cisco-developed enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

Note To take advantage of automated certificate and key rollover functionality, you must be running a CA that supports rollover and SCEP must be used as your client enrollment method. If you are running a Cisco IOS CA, you must be running Cisco IOS Release 12.4(2)T or a later release for rollover support.

- PKCS12--The router imports certificates in PKCS12 format from an external server.
- IOS File System (IFS)--The router uses any file system that is supported by Cisco IOS software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.

Note Prior to Cisco IOS Release 12.3(4)T, only the TFTP file system was supported within IFS.

- Manual cut-and-paste--The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the console terminal. A user may manually cut-and-paste certificate requests and certificates when there is no network connection between the router and CA.
- Enrollment profiles-- Enrollment profiles are primarily used for EST or terminal based enrollment. In case that the CA server does not support SCEP, the recommended methods for enrollment are EST based enrollment or terminal based enrollment.

- Self-signed certificate enrollment for a trustpoint--The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router's startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

Note To take advantage of autoenrollment and autoreenrollment, do not use either TFTP or manual cut-and-paste enrollment as your enrollment method. Both TFTP and manual cut-and-paste enrollment methods are manual enrollment processes, requiring user input.

Cisco IOS Suite-B Support for Certificate Enrollment for a PKI

Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B adds the following support for the certificate enrollment for a PKI:

- Elliptic Curve Digital Signature Algorithm (ECDSA) (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- PKI support for validation of for X.509 certificates using ECDSA signatures.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.

See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.

Registration Authorities

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA can be configured to automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

Automatic Certificate Enrollment

Automatic certificate enrollment allows the CA client to automatically request a certificate from its CA sever. This automatic router request eliminates the need for operator intervention when the enrollment request is sent to the CA server. Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. When the certificate expires, a new certificate is automatically requested.

Note When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

Automated Client Certificate and Key Rollover

By default, the automatic certificate enrollment function requests a new client certificate and keys from the CS before the client's current certificate expires. Certificate and key rollover allows the certificate renewal rollover request to be made before the certificate expires by retaining the current key and certificate until the new, or rollover, certificate is available. After a specified amount of time, the rollover certificate and keys will become the active certificate and keys. The expired certificate and keys are immediately deleted upon rollover and removed from the certificate chain and CRL.

The setup for automatic rollover is twofold: CA clients must be automatically enrolled and the client's CAs must be automatically enrolled and have the **auto-rollover** command enabled. For more information on configuring your CA servers for automatic certificate rollover see the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " of the *Public Key Infrastructure Configuration Guide*.

An optional renewal percentage parameter can be used with the **auto-enroll** command to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. In order for automatic rollover to occur, the renewal percentage must be less than 100.The specified percent value must not be less than 10. If a client certificate is issued for less than the configured validity period due to the impending expiration of the CA certificate, the rollover certificate will be issued for the balance of that period. A minimum of 10 percent of the configured validity period, with an absolute minimum of 3 minutes, is required to allow rollover enough time to function.

Tip If CA autoenrollment is not enabled, you may manually initiate rollover on an existing client with the **crypto pki enroll** command if the expiration time of the current client certificate is equal to or greater than the expiration time of the corresponding CA certificate. The client will initiate the rollover process, which occurs only if the server is configured for automated rollover and has an available rollover server certificate.

Note A key pair is also sent if configured by the **auto-enroll re-generate** command and keyword. It is recommended that a new key pair be issued for security reasons.

Certificate Enrollment Profiles

Certificate enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the **authentication url** command) and enrollment can be performed manually (using the **enrollment terminal** command).

Prior to Cisco IOS Release 12.3(11)T, certificate requests could be sent only in a PKCS10 format; however, an additional parameter was added to the profile, allowing users to specify the PKCS7 format for certificate renewal requests.

Note A single enrollment profile can have up to three separate sections for each task--certificate authentication, enrollment, and reenrollment.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. If you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure TFTP or manual cut-and-paste certificate enrollment, you cannot configure autoenrollment, autoreenrollment, an enrollment profile, nor can you utilize the automated CA certificate rollover capability.

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment or autoenrollment for clients participating in your PKI.

Before you begin

Before configuring automatic certificate enrollment requests, you should ensure that all necessary enrollment information is configured.

Prerequisites for Enabling Automated Client Certificate and Key Rollover

CA client support for certificate rollover is automatically enabled when using autoenrollment. For automatic CA certificate rollover to run successfully, the following prerequisites are applicable:

- Your network devices must support shadow PKI.
- Your clients must be running Cisco IOS Release 12.4(2)T or a later release.
- The client's CS must support automatic rollover. See the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " of the *Public Key Infrastructure Configuration Guide* for more information on CA server automatic rollover configuration.

Prerequisites for Specifying Autoenrollment Initial Key Generation Location

To specify the location of the autoenrollment initial key generation, you must be running Cisco IOS Release 12.4(11)T or a later release.

RSA Key Pair Restriction for Autoenrollment

Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own

key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatches.

Certificate renewal with regenerate option does not work with key label starting from zero ('0'), for example, '0test'. CLI allows configuring such name under trustpoint, and allows hostname starting from zero, but certificate regenerate will fail.

Restrictions for Automated Client Certificate and Key Rollover

In order for clients to run automatic CA certificate rollover successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.
- If the configuration cannot be saved to the startup configuration after a shadow certificate is generated, rollover will not occur.
- Rollover with key regenerate does not work when keypair name starts from zero ('0') (for example, '0test'). When configuring **rsakeypair** *name* under a trustpoint, do not configure name starting from zero. When keypair name is not configured and the default keypair is used, make sure the router hostname does not start from zero. If it does so, configure "**rsakeypair** *name* explicitly under the trustpoint with a different name.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode** | **retry period** *minutes* | **retry count** *number*] **url** *url* [**pem**]
5. **eckeypair** *label*
6. **subject-name** [*x.500-name*]
7. **vrf** *vrf-name*
8. **ip-address** {*ip-address* | *interface* | **none**}
9. serial-number [none]
10. **auto-enroll** [*percent*] [**regenerate**]
11. **usage** *method1* [*method2* [*method3*]]
12. **password** *string*
13. **rsakeypair** *key-label key-size encryption-key-size*]]
14. **fingerprint** *ca-fingerprint*
15. **on** *devicename* **:**
16. **exit**
17. **crypto pki authenticate** *name*

18. **exit**
19. **copy system:running-config nvram:startup-config**
20. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment** [**mode** | **retry period** *minutes* | **retry count** *number*] **url** *url* [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url http://cat.example.com`	Specifies the URL of the CA on which your router should send certificate requests. • **mode** --Specifies RA mode if your CA system provides an RA. • **retry period** *minutes* --Specifies the wait period between certificate request retries. The default is 1 minute between retries. • **retry count** *number* -- Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • **url** *url* -- URL of the file system where your router should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. • **pem** -- Adds privacy-enhanced mail (PEM) boundaries to the certificate request. **Note** An enrollment method other than TFTP or manual cut-and-paste must be configured to support autoenrollment.
Step 5	**eckeypair** *label* **Example:**	(Optional) Configures the trustpoint to use an Elliptic Curve (EC) key on which certificate requests are generated

	Command or Action	Purpose
	`Router(ca-trustpoint)# eckeypair Router_1_Key`	using ECDSA signatures. The *label* argument specifies the EC key label that is configured using the **crypto key generate rsa** or **crypto key generate ec keysize** command in global configuration mode. See the Configuring Internet Key Exchange for IPsec VPNs feature module for more information. **Note** If an ECDSA signed certificate is imported without a trustpoint configuration, then the label defaults to the FQDN value.
Step 6	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name cat`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name* --If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.
Step 7	**vrf** *vrf-name* **Example:** `Router(ca-trustpoint)# vrf myvrf`	(Optional) Specifies the the VRF instance in the public key infrastructure (PKI) trustpoint to be used for enrollment, certificate revocation list (CRL) retrieval, and online certificate status protocol (OCSP) status.
Step 8	**ip-address** {*ip-address* | *interface* | **none**} **Example:** `Router(ca-trustpoint)# ip address 192.168.1.66`	(Optional) Includes the IP address of the specified interface in the certificate request. • Issue the *ip-address* argument to specify either an IPv4 or IPv6 address. • Issue the *interface* argument to specify an interface on the router. • Issue the **none** keyword if no IP address should be included. **Note** If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 9	serial-number [none] **Example:** `Router(ca-trustpoint)# serial-number`	(Optional) Specifies the router serial number in the certificate request, unless the **none** keyword is issued. • Issue the **none** keyword to specify that a serial number will not be included in the certificate request.
Step 10	**auto-enroll** [*percent*] [**regenerate**] **Example:** `Router(ca-trustpoint)# auto-enroll regenerate`	(Optional) Enables autoenrollment, allowing the client to automatically request a rollover certificate from the CA. • If autoenrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration.

	Command or Action	Purpose
		• By default, only t he Domain Name System (DNS) name of the router is included in the certificate. • Use the *percent* argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. • Use the **regenerate** keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: "! RSA key pair associated with trustpoint is exportable." **Note** It is recommended that a new key pair be generated for security reasons.
Step 11	**usage** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# usage ssl-client`	(Optional) Specifies the intended use for the certificate. • Available options are **ike**, **ssl-client**, and **ssl-server**; the default is **ike**.
Step 12	**password** *string* **Example:** `Router(ca-trustpoint)# password string1`	(Optional) Specifies the revocation password for the certificate. • If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint. **Note** When SCEP is used, this password can be used to authorize the certificate request--often via a one-time password or similar mechanism.
Step 13	**rsakeypair** *key-label key-size encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair key-label 2048 2048`	(Optional) Specifies which key pair to associate with the certificate. • A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is not recommended.

	Command or Action	Purpose
		Note If this command is not enabled, the FQDN key pair is used.
Step 14	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.
Step 15	**on** *devicename* **:** **Example:** `Router(ca-trustpoint)# on usbtoken0:`	(Optional) Specifies that RSA keys will be created on the specified device upon autoenrollment initial key generation. • Devices that may be specified include NVRAM, local disks, and Universal Serial Bus (USB) tokens. USB tokens may be used as cryptographic devices in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.
Step 16	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 17	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 18	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Step 19	**copy system:running-config nvram:startup-config**
	Example:
	`Router#`
	`copy system:running-config nvram:startup-config`
		(Optional) Copies the running configuration to the NVRAM startup configuration.
		Note Autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

Step 20	**show crypto pki certificates** **Example:**	(Optional) Displays information about your certificates, including any rollover certificates.

	Command or Action	Purpose
	`Router# show crypto pki certificates`	

Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the manual cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

PEM-Formatted Files for Certificate Enrollment Request

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their routers.

Restrictions for Manual Certificate Enrollment

SCEP Restriction

We do not recommend switching URLs if SCEP is used; that is, if the enrollment URL is “http://myca,” do not change the enrollment URL after getting the CA certificate and before enrolling the certificate. A user can switch between TFTP and manual cut-and-paste.

Key Regeneration Restriction

Do not regenerate the keys manually using the **crypto key generate** command; key regeneration will occur when the **crypto pki enroll**command is issued if the **regenerate** keyword is specified.

Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure cut-and-paste certificate enrollment. This task helps you to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal pem**
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal pem** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies the manual cut-and-paste certificate enrollment method. • The certificate request will be displayed on the console terminal so that it may be manually copied (or cut). • **pem** --Configures the trustpoint to generate PEM-formatted certificate requests to the console terminal.
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it.
Step 8	crypto pki enroll name **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and displays the request for copying and pasting into the certificate server. • You are prompted for enrollment information, such as whether to include the router FQDN and IP address

	Command or Action	Purpose
		in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. • The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 9	crypto pki import name certificate **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate manually at the console terminal (pasting). • The base-64 encoded certificate is accepted from the console terminal and inserted into the internal certificate database. **Note** You must enter this command twice if usage keys, a signature key, and an encryption key are used. The first time the command is entered, one of the certificates is pasted into the router. The second time the command is entered, the other certificate is pasted into the router. It does not matter which certificate is pasted first. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If this applies to the certificate authority you are using, import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

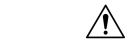
Configuring TFTP Certificate Enrollment

Perform this task to configure TFTP certificate enrollment. This task helps you to configure manual certificate enrollment using a TFTP server.

Before you begin

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.

- The router must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- If you are using a file specification with the **enrollment** command, the file must contain the CA certificate either in binary format or be base-64 encoded.
- You must know if your CA ignores key usage information in a certificate request and issues only a general purpose usage certificate.



Caution Some TFTP servers require that the file must exist on the server before it can be written. Most TFTP servers require files that can be written over. This requirement may pose a risk because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not be used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	**enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification`	Specifies TFTP as the enrollment method to send the enrollment request and to retrieve the CA certificate and router certificate and any optional parameters. **Note** For TFTP enrollment, the URL must be configured as a TFTP URL, tftp://example_tftp_url. • An optional file specification filename may be included in the TFTP URL. If the file specification is not included, the FQDN will be used. If the file specification is included, the router will append the extension ".ca" to the specified filename.
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it from the specified TFTP server.
Step 8	crypto pki enroll name **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and writes the request out to the TFTP server. • You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are queried about whether to display the certificate request to the console terminal. • The filename to be written is appended with the extension ".req". For usage keys, a signature key and an encryption key, two requests are generated and sent. The usage key request filenames are appended with the extensions "-sign.req" and "-encr.req", respectively.
Step 9	crypto pki import name certificate **Example:**	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate.

	Command or Action	Purpose
	`Router(config)# crypto pki import mytp certificate`	• The router will attempt to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. • The router will parse the received files, verify the certificates, and insert the certificates into the internal certificate database on the router. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Certifying a URL Link for Secure Communication with a Trend Micro Server

Perform this task to certify a link used in URL filtering that allows secure communication with a Trend Micro Server.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **clock set** *hh* **:** *mm* **:** *ss date month year*
3. **configure terminal**
4. **clock timezone** *zone hours-offset* [*minutes-offset*]
5. **ip http server**
6. **hostname** *name*
7. **ip domain-name** *name*

8. **crypto key generate rsa general-keys modulus** *modulus-size*
9. **crypto pki trustpoint** *name*
10. **enrollment terminal**
11. **crypto ca authenticate** *name*
12. Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.
13. Enter **yes** to accept this certificate.
14. **serial-number**
15. **revocation-check none**
16. **end**
17. **trm register**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clock set** *hh* **:** *mm* **:** *ss date month year* **Example:** `Router# clock set 23:22:00 22 Dec 2009`	Sets the clock on the router.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**clock timezone** *zone hours-offset* [*minutes-offset*] **Example:** `Router(config)# clock timezone PST -08`	Sets the time zone. • The *zone* argument is the name of the time zone (typically a standard acronym). The *hours-offset* argument is the number of hours the time zone is different from Universal Time Coordinated (UTC). The *minutes-offset* argument is the number of minutes the time zone is different from UTC. **Note** The *minutes-offset*argument of the **clock timezone** command is available for those cases where a local time zone is a percentage of an hour different from UTC or Greenwich Mean Time (GMT). For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5. In this case, the necessary command would be **clock timezone AST -3 30**.

	Command or Action	Purpose
Step 5	**ip http server** **Example:** `Router(config)# ip http server`	Enables the HTTP server.
Step 6	**hostname** *name* **Example:** `Router(config)# hostname hostname1`	Configures the hostname of the router.
Step 7	**ip domain-name** *name* **Example:** `Router(config)# ip domain-name example.com`	Defines the domain name for the router.
Step 8	**crypto key generate rsa general-keys modulus** *modulus-size* **Example:** `Router(config)# crypto key generate rsa general-keys modulus general`	Generates the crypto keys. • The **general-keys** keyword specifies that a general purpose key pair is generated, which is the default. • The **modulus** keyword and *modulus-size* argument specify the IP size of the key modulus. By default, the modulus of a CA key is 1024 bits. When generating RSA keys, you will be prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate and to use. A length of less than 2048 is not recommended. **Note** The name for the general keys that are generated are based on the domain name that is configured in Step 7. For example, the keys will be called "example.com."
Step 9	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 10	**enrollment terminal** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies the manual cut-and-paste certificate enrollment method. • The certificate request will be displayed on the console terminal so that you may manually copy (or cut).
Step 11	**crypto ca authenticate** *name* **Example:**	Takes the name of the CA as the argument and authenticates it.

	Command or Action	Purpose

	`Router(ca-trustpoint)# crypto ca authenticate mytp`	• The following command output displays:
		`Enter the base 64 encoded CA certificate.`
		`End with a blank line or the word "quit" on a line by itself.`

Step 12	Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.	`MIIDIDCCAcmgAwIBAgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJV`
		`UzEQMA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBTZWN1cmUgQ2Vy`
		`dGlmaWNhdGUgQXV0aG9yaXR5MB4XDTk4MDgyMjE2NDE1MVoXDTE4MDgyMjE2NDE1`
		`MVowTjELMAkGA1UEBhMCVVMxEDAOBgNVBAoTB0VxdWlmYXgxLTArBgNVBAsTJEVx`
		`dWlmYXggU2VjdXJlIENlcnRpZmljYXRlIEF1dGhvcml0eTCBnzANBgkqhkiG9w0B`
		`AQEFAAOBjQAwgYkCgYEAwV2xWGcIYu6gmi0fCG2RFGiYCh7+2gRvE4RiIcPRfM6f`
		`BeC4AfBONOziipUEZKzxa1NfBbPLZ4C/QgKO/t0BCezhABRP/PvwDN1Dulsr4R+A`
		`cJkVV5MW8Q+XarfCaCMczE1ZMKxRHjuvK9buY0V7xdlfUNLjUA86iOe/FP3gx7kC`
		`AwEAAaOCAQkwggEFMHAGA1UdHwRpMGcwZaBjoGGkXzBdMQswCQYDVQQGEwJVUzEQ`
		`MA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBTZWN1cmUgQ2VydGlm`
		`aWNhdGUgQXV0aG9yaXR5MQ0wCwYDVQQDEwRDUkwxMBoGA1UdEAQTMBGBDzIwMTgw`
		`ODIyMTY0MTUxWjALBgNVHQ8EBAMCAQYwHwYDVR0jBBgwFoAUSOZo+SvSspXXR9gj`
		`IBBPM5iQn9QwHQYDVR0OBBYEFEjmaPkr0rKV10fYIyAQTzOYkJ/UMAwGA1UdEwQF`
		`MAMBAf8wGgYJKoZIhvZ9B0FABA0wCxsFVjMuMGMDAgbAMA0GCSqGSIb3DQEBBQUA`
		`A4GBAFjOKer89961zgK5F7WF0bnj4JXMJTENAKaSbn+2kmOeUJXRmm/kEd5jhW6Y`
		`7qj/WsjTVbJmcVfewCHrPSqnI0kBBIZCe/zuf6IWUrVnZ9NA2zsmWLIodz2uFHdh`
		`1voqZiegDfqnc1zqcPGUIWVEX/r87yloqaKHee9570+sB3c4`
		The following command output displays:
		`Certificate has the following attributes:`
		`Fingerprint MD5: 67CB9DC0 13248A82 9BB2171E D11BECD4`

	Command or Action	Purpose
		`Fingerprint SHA1: D23209AD 23D31423 2174E40D 7F9D6213 9786633A`

Step 13	Enter **yes** to accept this certificate.	`% Do you accept this certificate? [yes/no]: yes`
		The following command output displays:
		`Trustpoint CA certificate accepted.`
		`% Certificate successfully imported`

Step 14	**serial-number** **Example:** `hostname1(ca-trustpoint)# serial-number`	Specifies the router serial number in the certificate request.
Step 15	**revocation-check none** **Example:** `hostname1(ca-trustpoint)# revocation-check none` **Example:**	Specifies that certificate checking is ignored.
Step 16	**end** **Example:** `hostname1(ca-trustpoint)# end`	Exits ca-trustpoint configuration mode and returns to privileged EXEC mode.
Step 17	**trm register** **Example:** `hostname1# trm register`	Manually starts the Trend Micro Server registration process.

Configuring a Persistent Self-Signed Certificate for Enrollment via SSL

This section contains the following tasks:

Note These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

Persistent Self-Signed Certificates Overview

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, the client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a PKI application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate, so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router's startup configuration.

Restrictions

- You can configure only one trustpoint for a persistent self-signed certificate.
- The maximum lifetime of a self-signed certificate is 00:00:00 GMT Jan 1, 2030.

Note Do not change the IP domain name or the hostname of the router after creating the self-signed certificate. Changing either name triggers the regeneration of the self-signed certificate and overrides the configured trustpoint. WebVPN ties the SSL trustpoint name to the WebVPN gateway configuration. If a new self-signed certificate is triggered, then the new trustpoint name does not match the WebVPN configuration, causing the WebVPN connections to fail.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** name
4. **enrollment selfsigned**
5. **subject-name** [*x.500-name*]

6. **rsakeypair** *key-label* [key-size [encryption-key-size]]
7. **crypto pki enroll** name
8. **end**
9. **show crypto pki certificates** [*trustpoint-name*[**verbose**]]
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** name **Example:** `Router(config)# crypto pki trustpoint local`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 4	**enrollment selfsigned** **Example:** `Router(ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name`	(Optional) Specifies the requested subject name to be used in the certificate request. • If no value for the *x-500-name* argument is specified, the FQDN, which is the default subject name, is used.
Step 6	**rsakeypair** *key-label* [key-size [encryption-key-size]] **Example:** `Router(ca-trustpoint)# rsakeypair examplekey 2048`	(Optional) Specifies which key pair to associate with the certificate. • The value for the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify a value for the *key-size* argument for generating the key, and specify a value for the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is no recommended.

	Command or Action	Purpose
		Note If this command is not enabled, the FQDN key pair is used.
Step 7	**crypto pki enroll** name **Example:** `Router(config)# crypto pki enroll local`	Tells the router to generate the persistent self-signed certificate.
Step 8	**end** **Example:** `Router(ca-trustpoint)# end`	(Optional) Exits ca-trustpoint configuration mode. • Enter this command a second time to exit global configuration mode.
Step 9	**show crypto pki certificates** [*trustpoint-name*[**verbose**]] **Example:** `Router# show crypto pki certificates local verbose`	Displays information about your certificate, the certification authority certificate, and any registration authority certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints status`	Displays the trustpoints that are configured in the router.

Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.

Before you begin

To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as the server is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http secure-server**
4. **end**
5. **copy system:running-config nvram: startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http secure-server** **Example:** `Router(config)# ip http secure-server`	Enables the HTTPS web server. **Note** A key pair (modulus 1024) and a self-signed certificate are automatically generated.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.
Step 5	**copy system:running-config nvram: startup-config** **Example:** `Router# copy system:running-config nvram: startup-config`	Saves the self-signed certificate and the HTTPS server in enabled mode.

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure a certificate enrollment profile for enrollment or reenrollment. This task helps you to configure an enrollment profile for certificate enrollment or reenrollment of a router with a Cisco IOS CA that is already enrolled with a third-party vendor CA.

Enable a router that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted. To enable this functionality, you must issue the **enrollment credential** command. Also, you cannot configure manual certificate enrollment.

Before you begin

Perform the following tasks at the client router before configuring a certificate enrollment profile for the client router that is already enrolled with a third-party vendor CA so that the router can reenroll with a Cisco IOS certificate server:

- Defined a trustpoint that points to the third-party vendor CA.
- Authenticated and enrolled the client router with the third-party vendor CA.

Note

- To use certificate profiles, your network must have an HTTP interface to the CA.
- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. enrollment profile label
5. **exit**
6. **crypto pki profile enrollment** *label*
7. Do one of the following:
   - **authentication url** *url*
   - **authentication terminal**
8. **authentication command**
9. Do one of the following:
   - **enrollment url** *url*
   -
   - **enrollment terminal**
10. **enrollment credential** *label*
11. **enrollment command**
12. **parameter** *number* {**value** *value* | **prompt** *string*}
13. **exit**
14. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Entrust`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	enrollment profile label **Example:** `Router(ca-trustpoint)# enrollment profile E`	Specifies that an enrollment profile is to be used for certificate authentication and enrollment.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki profile enrollment** *label* **Example:** `Router(config)# crypto pki profile enrollment E`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label* --Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.

Step 7	Do one of the following:
	• **authentication url** *url*
	• **authentication terminal**
	Example:
	`Router(ca-profile-enroll)# authentication url http://entrust:81`
	Example:
	`Router(ca-profile-enroll)# authentication terminal`
		Specifies the URL of the CA server to which to send certificate authentication requests.
		• *url* --URL of the CA server to which your router should send authentication requests. If you are using HTTP, the URL should read "http://CA_name," where CA_name is the host DNS name or IP address of the CA. If you are using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the FQDN of the router will be used.)
		Specifies manual cut-and-paste certificate authentication.

Step 8	**authentication command** **Example:** `Router(ca-profile-enroll)# authentication command`	(Optional) Specifies the HTTP command that is sent to the CA for authentication.
Step 9	Do one of the following: • **enrollment url** *url* • • **enrollment terminal** **Example:**	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. Specifies manual cut-and-paste certificate enrollment.

	Command or Action	Purpose

	`Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe`
	Example:
	Example:
	`Router(ca-profile-enroll)# enrollment terminal`	

Step 10	**enrollment credential** *label* **Example:** `Router(ca-profile-enroll)# enrollment credential Entrust`	(Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS CA. **Note** This command cannot be issued if manual certificate enrollment is being used.
Step 11	**enrollment command** **Example:** `Router(ca-profile-enroll)# enrollment command`	(Optional) Specifies the HTTP command that is sent to the CA for enrollment.
Step 12	**parameter** *number* {**value** *value* | **prompt** *string*} **Example:** `Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc`	(Optional) Specifies parameters for an enrollment profile. • This command can be used multiple times to specify multiple values.
Step 13	**exit** **Example:** `Router(ca-profile-enroll)# exit`	(Optional) Exits ca-profile-enroll configuration mode. • Enter this command a second time to exit global configuration mode.
Step 14	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

What to Do Next

If you configured the router to reenroll with a Cisco IOS CA, you should configure the Cisco IOS certificate server to accept enrollment requests only from clients already enrolled with the specified third-party vendor CA trustpoint to take advantage of this functionality. For more information, see the module " Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment."

Configuring Certificate Enrollment in a Two-Tier PKI Environment

The feature enables sub-CAs to issue certificates to their clients when a root CA is offline. The root certificate can be imported through the CLI first, and then it is used to validate the issuing sub CA certificate configured under the trustpoint.

Note Enable revocation checking as per your environment before performing the following tasks.

For importing the ROOT-CA through terminal, perform the following steps:

```
enable
!
configure terminal
!
crypto pki trustpoint ROOT-CA
revocation-check none
enrollment terminal
!
crypto pki authenticate ROOT-CA
!
exit
```

For authenticating SUB-CA without specifying or accepting the fingerprint.

```
enable
!
configure terminal
!
crypto pki trustpoint SUB-CA
revocation-check none
enrollment url url
chain-validation continue ROOT-CA
exit
!
crypto pki authenticate SUB-CA
exit
```

Configuring Certificate Renewal by Enabling Multiple Trustpoints

Starting from the Cisco IOS XE 17.4.1 release, you can enable the registration authority to use multiple trustpoints to validate router credentials for initial certificate enrollment and certificate renewal. This enhancement enables automated validation of multiple trustpoints while maintaining zero-touch certificate enrollment through the SCEP enrollment protocol.

When you enroll a router for the first time, an SCEP request is initiated and this request is signed by using the SUDI credentials. The request is then sent to a registration authority which validates the SUDI certificate through a local trustpoint. The local trustpoint validates the router SCEP credentials. If the validation is successful, the registration authority uses the SUDI certificate to decrypt the signature and validate the hash. After the hash validation is also successful, the registration authority forwards the SCEP request to the certificate authority (CA). The CA then signs the request and sends the certificate back to the registration authority which in turn forwards the certificate to the router. At this point, the SCEP enrollment is complete.

In the case of a certificate renewal, when the same process is followed, the renewal fails. This is because the registration authority cannot validate the renewal request since the router uses the current certificate as the

credentials. Since the registration authority can use only one trustpoint to validate the router identity, the certificate renewal fails.

To overcome this challenge, you can now configure the registration authority to use multiple trustpoints to validate the router credentials. In this manner, the initial enrollment as well as the renewal works seamlessly.

To configure multiple trustpoints, use the **grant auto <tp-list>** command. You can configure from upto 5 trustpoints by using this command. For example:

```
grant auto tp-list <tp1 tp2>
grant auto tp-list <tp1 tp2 tp3>
grant auto tp-list <tp1 tp2 tp3 tp4>
grant auto tp-list <tp1 tp2 tp3 tp4 tp5>
```

After you configure the trustpoints, the registration authority validates the certificates that are received by using one of the configured trustpoints. The validation starts from the first trustpoint. If the validation is successful, the certificate is renewed. Else, the authority validates using the next available trustpoint.

Sample Configuration

```
crypto pki server FANRSACA
no database archive
grant auto <tp-list> ACT2_SUDI_CA <CA_TRUSTPOINT>
hash sha256
mode ra transparent
!
crypto pki trustpoint FANRSACA
enrollment url http://10.4.1.117:8080/ejbca/publicweb/apply/scep/FANRSACA
serial-number none
fqdn none
ip-address none
subject-name serialNumber=PID:ISR4451-X/K9 SN:FOC23231CRY, CN=ISR4k-1-ra
revocation-check none
rsakeypair FANRSACA_Key 4096
!
crypto pki trustpoint ACT2_SUDI_CA
enrollment profile ACT2_SUDI_CA
revocation-check none
!
crypto pki trustpool policy
revocation-check none
```

Note **Grant auto trustpoint** and **grant auto tp-list** are mutually exclusive. You cannot run the **grant auto tp-list** command if you have already configured grant auto trustpoint.

Configuration Examples for PKI Certificate Enrollment Requests

Configuring Certificate Enrollment or Autoenrollment Example

The following example shows the configuration for the “mytp-A” certificate server and its associated trustpoint, where RSA keys generated by the initial autoenrollment for the trustpoint will be stored on a USB token, “usbtoken0”:

```
crypto pki server mytp-A
```

```
   database level complete
   issuer-name CN=company, L=city, C=country
   grant auto
! Specifies that certificate requests will be granted automatically.
!
crypto pki trustpoint mytp-A
   revocation-check none
   rsakeypair myTP-A
   storage usbtoken0:
! Specifies that keys will be stored on usbtoken0:.
   on usbtoken0:
```

! Specifies that keys generated on initial auto enroll will be generated on and stored o**n !** usbtoken0:

Configuring Autoenrollment Example

The following example shows how to configure the router to automatically enroll with a CA on startup, enabling automatic rollover, and how to specify all necessary enrollment information in the configuration:

```
crypto pki trustpoint trustpt1
 enrollment url http://trustpt1.example.com//
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet-0
 serial-number none
 usage ike
 auto-enroll regenerate
 password password1
 rsa-key trustpt1 2048
!
crypto pki certificate chain trustpt1
certificate pki 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Note In this example, keys are neither regenerated nor rolled over.

Configuring Certificate Autoenrollment with Key Regeneration Example

The following example shows how to configure the router to automatically enroll with the CA named "trustme1" on startup and enable automatic rollover. The **regenerate** keyword is issued, so a new key will be generated for the certificate and reissued when the automatic rollover process is initiated. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
crypto pki trustpoint trustme1
 enrollment url http://trustme1.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 auto-enroll 90 regenerate
 password password1
 rsakeypair trustme1 2048
 exit
crypto pki authenticate trustme1
copy system:running-config nvram:startup-config
```

Configuring Cut-and-Paste Certificate Enrollment Example

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method:

```
Router(config)#
crypto pki trustpoint TP
Router(ca-trustpoint)#
enrollment terminal
Router(ca-trustpoint)#
crypto pki authenticate TP
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
-----BEGIN CERTIFICATE-----
MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==
-----END CERTIFICATE-----
Certificate has the following attributes:
Fingerprint: D6C12961 CD78808A 4E02193C 0790082A
% Do you accept this certificate? [yes/no]:
y
Trustpoint CA certificate accepted.
% Certificate successfully imported
Router(config)#
crypto pki enroll TP
% Start certificate enrollment..
```

```
% The subject name in the certificate will be:
Router.example.com
% Include the router serial number in the subject name? [yes/no]:
n
% Include an IP address in the subject name? [no]:
n
Display Certificate Request to terminal? [yes/no]:
y
Signature key certificate request -
Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+
!
!
!
Redisplay enrollment request? [yes/no]:
Encryption key certificate request -
Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
eNPFhozcaQ/2
!
!
!
Redisplay enrollment request? [yes/no]:
n
Router(config)#
crypto pki import TP certificate
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=
% Router Certificate successfully imported
Router(config)#
```

```
crypto pki import TP cert
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=
% Router Certificate successfully imported
```

You can verify that the certificate was successfully imported by issuing the **show crypto pki certificates** command:

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 14DECE05000000000C48
  Certificate Usage: Encryption
  Issuer:
    CN = TPCA-root
     O = Company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:45 PDT Jun 7 2002
    end   date: 18:26:45 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP
Certificate
  Status: Available
  Certificate Serial Number: 14DEC2E9000000000C47
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:42 PDT Jun 7 2002
    end   date: 18:26:42 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
```

```
  Associated Trustpoints: TP
CA Certificate
  Status: Available
  Certificate Serial Number: 3AC0A65E9547C2874AAF2468A942D5EE
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = Company
     C = US
  Subject:
    CN = tpca-root
     O = company
     C = US
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 16:46:01 PST Feb 13 2002
    end   date: 16:54:48 PST Feb 13 2007
  Associated Trustpoints: TP
```

Configuring Manual Certificate Enrollment with Key Regeneration Example

The following example shows how to regenerate new keys with a manual certificate enrollment from the CA named “trustme2”:

```
crypto pki trustpoint trustme2
 enrollment url http://trustme2.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 regenerate
 password password1
 rsakeypair trustme2 2048
 exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
```

Creating and Verifying a Persistent Self-Signed Certificate Example

The following example shows how to declare and enroll a trustpoint named “local” and generate a self-signed certificate with an IP address:

```
crypto pki trustpoint local
 enrollment selfsigned
 end
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

```
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ip http secure-server
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified.  Issue "write memory"
to save new certificate
Router(config)#
```

Note You need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your Access Control Lists (ACLs) to permit or deny SSH access to the router. You can use the **ip ssh rsa keypair-name** *unexisting-key-pair-name* command to disable the SSH server.

Verifying the Self-Signed Certificate Configuration Example

The following example displays information about the self-signed certificate that you just created:

```
Router# show crypto pki certificates
Router Self-Signed Certificate
  Status: Available
  Certificate Serial Number: 01
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
  Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
  Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
  Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 is the router's serial number and varies depending on the router's actual serial number.

The following example displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto key mypubkey rsa
% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
  6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
  BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
  6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
  2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
  463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
  8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
  34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named "local":

```
Router# show crypto pki trustpoints
Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.example.com
          Serial Number: 01
    Persistent self-signed certificate trust point
```

Configuring Direct HTTP Enrollment Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
crypto pki trustpoint Entrust
 enrollment profile E
 serial
crypto pki profile enrollment E
 authentication url http://entrust:81
 authentication command GET /certs/cacert.der
 enrollment url http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command POST reference_number=$P2&authcode=$P1
 &retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
```

```
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Configuring Certificate Enrollment in a Two-Tier PKI Environment Example

Example of importing the ROOT-CA via terminal.

```
(config)#crypto pki trustpoint ROOT-CA
(ca-trustpoint)#revocation-check none
(ca-trustpoint)#enrollment terminal


(config)#crypto pki authenticate ROOT-CA

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgIQIfTArEE1yKZPXHaAVgDk5jANBgkqhkiG9w0BAQsFADBN
MRMwEQYKCZImiZPyLGQBGRYDY29tMRgwFgYKCZImiZPyLGQBGRYIdnBuLWVhc3Qx
HDAaBgNVBAMTE3Zwbi1lYXN0LXphY2ttY2ktQ0EwHhcNMTgxMjIwMDAwNjMyWhcN
MjgxMjIwMDAxNjMyWjBNMRMwEQYKCZImiZPyLGQBGRYDY29tMRgwFgYKCZImiZPy
LGQBGRYIdnBuLWVhc3QxHDAaBgNVBAMTE3Zwbi1lYXN0LXphY2ttY2ktQ0EwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC9Gdns9lU2HHc+XYhrmZKg6+Xo
5kNflu6mMgCfZ7ZiAKxZ03whJWZqNC7JRZQ+LkIJAcBUSf2mSJWRp+HVgI6k4Zf7
bMgIBq629HT8XmFLrr3lfh1lfL7WqI1Uez7/PEzjsw09y/m/WiSnrlgR3+PvyDbH
E86A6JnmtTNIs4qawUe72BlnEzwwRaFNi7VQz7GQw3CUo+RX9wtFYjABTyTUM/BA
MP47pI8CVh1jHVHqHcbqpyd97j1/8n1d/NCmcHKIg2hnKEO1Hx8oK7QIHe1rkryl
+r0ol2fS3CGgY000+FINs3qw4h8H8xfmsc5cs8lJCIbZGJhMTXq6u4Ecp+N1AgMB
AAGjUTBPMAsGA1UdDwQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTb
zvfa7aNZspz3GwJCvKDIKO8KFTAQBgkrBgEEAYI3FQEEAwIBADANBgkqhkiG9w0B
AQsFAAOCAQEAgTIPTauHsPp7h1v/iFXkbVV1aG7O8/IaJG0sCr0f9/nsfM9HO0Jm
LP+twy5KkFa7I6u4vMlMlfNyujS60Fqnw3m8UJCy2SkYVwlGrBddN+BQbnkZ460M
sYfaynFBsvsbmmaLEqUQ3t9cmNCskXoda+FffyFTwAUBFzV66BGKpn6Y7oyIghF5
NLjjgWPVmRy7RKM4IKe9J0+oEmnugwtdfHgiFdX+d6qPovjbApj2j6N4+Cv6qHDO
/c+wUXRxz08eFNOqHNJipk70OXMrUh4UaWMnM/CYA9E1sjjSAWhBl4ii/+fiaILw
xgof+2mmIzafzFZz+eVf5kgwpV07GlZlng==
-----END CERTIFICATE-----
quit
Certificate has the following attributes:
       Fingerprint MD5: 99182E1E 96FB0595 DF86BFCE 3C781CF5
      Fingerprint SHA1: 6E55B878 9AA3B603 D689AC25 F027615E 0C88E6E4

% Do you accept this certificate? [yes/no]: yes

Authenticating SUB-CA without having to specify or accept the fingerprint.

(config)#crypto pki trustpoint SUB-CA
(ca-trustpoint)#enrollment url  http://<SUBCA_IP/FQDN>:80/certsrv/mscep/mscep.dll
(ca-trustpoint)#chain-validation continue ROOT-CA
(ca-trustpoint)#revocation-check none

(ca-trustpoint)#crypto pki authenticate SUB-CA
Certificate has the following attributes:
       Fingerprint MD5: 5C38CB0A 050AAE87 84A08A75 5F7084B8
      Fingerprint SHA1: EB829470 B8B9E26E 4457F346 7A3E957C C623C6F9
Certificate validated - Signed by existing trustpoint CA certificate.

Trustpoint CA certificate accepted.
```

Additional References

Related Documents

Related Topic	Document Title
USB token RSA operations: Benefits of using USB tokens	“Storing PKI Credentials” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
USB token RSA operations: Certificate server configuration	“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment” chapter in the Cisco IOS Security Configuration Guide: Secure Connectivity See the “Generating a Certificate Server RSA Key Pair” section, the “Configuring a Certificate Server Trustpoint” section, and related examples.
Overview of PKI, including RSA keys, certificate enrollment, and CAs	“ Cisco IOS PKI Overview: Understanding and Planning a PKI ” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Secure Device Provisioning: functionality overview and configuration tasks	“ Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI ” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
RSA key generation and deployment	“ Deploying RSA Keys Within a PKI ” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Cisco IOS certificate server overview information and configuration tasks	“ Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment ” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Setting up and using a USB token	“ Storing PKI Credentials ” module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Cisco IOS security commands	*Cisco IOS Security Command Reference*
Suite-B ESP transforms	Configuring Security for VPNs with IPsec feature module.
Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration.	Configuring Internet Key Exchange for IPsec VPNs feature module.
Suite-B Integrity algorithm type transform configuration.	Configuring Internet Key Exchange Version 2 (IKEv2) feature module.
Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2.	Configuring Internet Key Exchange Version 2 (IKEv2) feature module.

Related Topic	Document Title
Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	Configuring Internet Key Exchange for IPsec VPNs and Configuring Internet Key Exchange Version 2 (IKEv2) feature modules.
Recommended cryptographic algorithms	*Next Generation Encryption*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 149: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 112

Setting Up Secure Device Provisioning for Enrollment in a PKI

This module describes how to use Secure Device Provisioning (SDP) in a public key infrastructure (PKI). SDP is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. The end devices may or may not be directly connected to the network at the time of deployment or provisioning. SDP provides a solution for users deploying a large number of peer devices (including certificates and configurations).

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

Setting Up SDP for Enrollment in a PKI

Before you set up SDP, your environment should meet the following requirements:

- The petitioner device and the server must have IP connectivity between each other.
- The introducer must have a web browser that supports JavaScript.
- The introducer must have enable privileges on the client device.

- A Cisco IOS Release 12.3(8)T PKI-enabled image or a later image.

Setting Up SDP for Enrollment in a PKI Using USB Tokens

To leverage USB tokens to provision devices with SDP, your environment should meet the following requirements:

- Both the petitioner device and the server must have IP connectivity between each other.
- The introducer must have a web browser that supports JavaScript.
- The introducer must have enable privileges on the client device.
- The introducer must have access to a petitioner device.
- The introducer must have access to the USB token and PIN, if configured.
- A Cisco IOS Release 12.4(15)T PKI-enabled image or a later image.

Note Cisco IOS Release 12.4(15)T or a later release provides the flexibility to move credentials stored on the USB token. However, the device used to configure the USB token may run any Cisco IOS Release 12.3(14)T PKI-enabled image or a later image.

Using SDP to Configure a Device for an Internet Connection Through a Service Provider

To leverage SDP to configure a device that is not connected to the Internet, your environment should meet the following requirements:

- The introducer must have a web browser that supports JavaScript.
- The introducer must have enable privileges on the client device.
- A Cisco router that supports a DHCP client and a PPPoE client and has a configured LAN or WAN interface.
- A Cisco IOS Release 12.4(20)T PKI-enabled image or a later image. If a previous Cisco IOS release is used on one of the devices, the SDP functionality defaults to the earlier Cisco IOS version.

Information About Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

SDP Overview

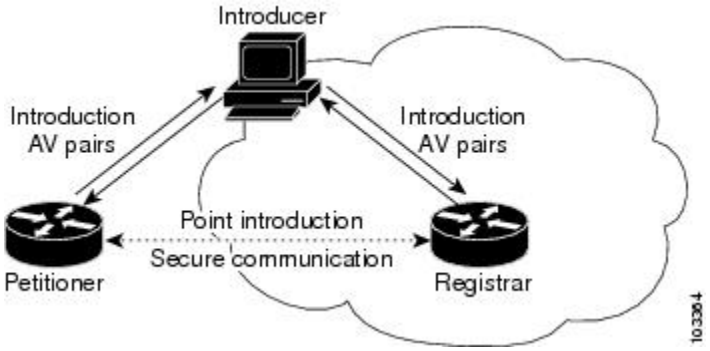
SDP (also refer red to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities (see the figure below):

- Introducer--A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.

- An introducer can be configured as an administrative introducer, which allows an administrator performing the introduction to supply the name for the device being introduced. The supplied device name is used as if it were the name of an introducer in the normal SDP mechanisms, preserving the existing functionality of the SDP configuration. For more information on function of the administrative introducer, see the section Authentication and Authorization Lists for an Administrative Introducer, on page 1251.

- Petitioner--A client, or new device, to be introduced to the secure network.
- Registrar--A server that authorizes the petitioner. The registrar can be a certificate server.

Figure 37: Post-Introduction Secure Communication



As of Cisco IOS Release 12.4(20)T or a later release, the introducer can start the SDP process without establishing prior Internet connectivity on the petitioner. The use of the prep-connect phase and the connect phase provides the ability to configure a petitioner for Internet connectivity through a service provider. See the How SDP Works, on page 1241 for more information on the prep-connect phase and the connect phase.

The registrar communicates directly with an external authentication, authorization, and accounting (AAA) server to verify petitioner credentials, permit or deny enrollment, and retrieve specific petitioner configuration information. The petitioner and registrar serve web pages to the introducer, the end user. The petitioner receives the bootstrap configuration from a remote management system through the introducer's web browser.

SDP is implemented over a web browser with six possible phases--prep-connect (optional), connect, start (optional), welcome, introduction, and completion. Each phase is shown to the user through a web page. See the How SDP Works, on page 1241 for more information on each phase.

How SDP Works

The following sections describe how SDP deploys PKI between two devices:

The SDP process starts with one of three entry pages being loaded into the web browser by the introducer: the SDP prep-connect phase received from the administrator; the start phase loaded from the registrar; or the welcome phase loaded from the petitioner.

The sample figures show how to introduce the local device (the petitioner) to the secure domain of the registrar. The “introducer” is referred to as the end user.

SDP Prep-Connect Phase

The prep-connect page is optional. Without the prep-connect page, the petitioner must have IP connectivity established.

The administrator must configure the prep-connect template and send the prep-connect page to the introducer. See the Default Prep-Connect Template, on page 1256 for more information.

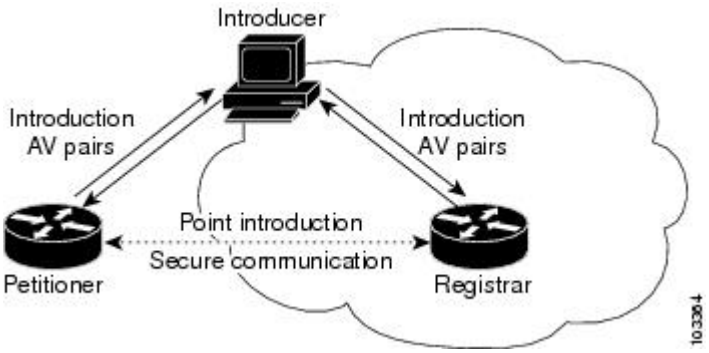
The administrator must also obtain and communicate the username and password for the secure network to the introducer by a telephone call, an e-mail, a secure e-mail, a CD, or a USB token. The registrar may be configured to authenticate the introducer using an existing AAA infrastructure (for example, an existing username and password database that is part of the existing corporate domain). The SDP prep-connect phase supports a challenge password mechanism as is used by common AAA infrastructures. See the How SDP Uses an External AAA Database, on page 1250 for more information.

After receiving the prep-connect page, the introducer must load the page onto the computer where the HTTP browser operates. The introducer then loads the prep-connect page into the HTTP browser as a local file and then the prep-connect page is displayed (see the figure below).

Figure 38: Sample SDP Prep-Connect Page

After the introducer clicks the Log onto Cisco Device button, the login dialog box is displayed (see the figure below). The introducer enters the factory default username (cisco) and password (cisco) of the Cisco device.

Figure 39: Sample Petitioner Login Dialog Box



The introducer authenticates with the petitioner and then Internet connectivity is tested by attempting to access a known URL. Access to www.cisco.com (198.133.219.25) is tested by default. The administrator can modify the URL to be used for testing connectivity by modifying the default prep-connect template. For more information about modifying the default test URL and other fields that the administrator may configure for the prep-connect page, see the section Default Prep-Connect Template, on page 1256.

Note To mitigate the possibility that the prep-connect page could be modified to contain an IP address of an untrusted registrar or that a prep-connect page might be e-mailed from an untrusted source, use a secure method, such as secure e-mail, to send the prep-connect page.

If Internet connectivity is established either the start page or welcome page is displayed, depending on the prep-connect template setting as defined by the administrator. If Internet connectivity is not established, the connect page is displayed.

SDP Connect Phase

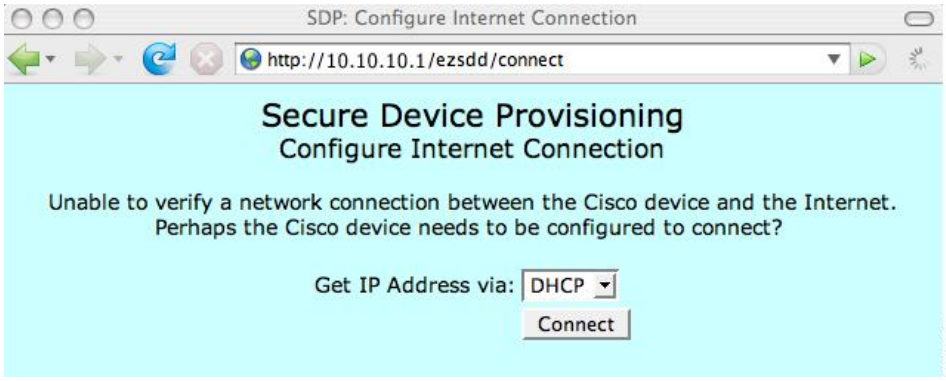
The connect page is displayed only if the prep-connect page is used and there is no IP connectivity for the petitioner at the completion of the prep-connect phase. The connect page has three IP address assignment methods to allow flexibility for your Cisco IOS platform: Dynamic Host Configuration Protocol (DHCP), Point to Point Protocol over Ethernet (PPPoE), or static IP address assignment.

Note SDP functionality is not used with the Cisco IOS configuration to establish Internet connectivity. SDP functionality includes a signature on the Cisco IOS configuration, guaranteeing that the values have not changed in transit.

DHCP IP Address Assignment Method

If the introducer chooses DHCP, the default method, for the IP address assignment method option (see the figure below), clicking the Connect button causes the petitioner to be configured for Internet connectivity.

Figure 40: Sample Connect Page for DHCP IP Address Assignment Method



PPPoE IP Address Assignment Method

If the introducer chooses PPPoE, input fields for PPPoE username and password are displayed (see the figure below). The introducer must enter the username and password as supplied by the Internet service provider (ISP) and then click the Connect button, which causes petitioner to be configured for Internet connectivity.

Figure 41: Sample Connect Page for PPPoE IP Address Assignment Method

Static IP Address Assignment Method

If the introducer chooses static, input fields for the IP address, netmask, and the default gateway are displayed (see the figure below). The introducer must enter the configuration values as supplied by the ISP and then click the Connect button, which causes petitioner to be configured for Internet connectivity.

Figure 42: Connect Page for Static IP Address Assignment Method

Connect Page IP Address Configuration

After IP address configuration, Internet connectivity is tested again by attempting to access a known URL configured by the administrator in the prep-connect template (www.cisco.com by default). If Internet connectivity is now established either the start page or welcome page is displayed, depending on the

prep-connect template setting as defined by the administrator. If Internet connectivity is not established, the introducer should verify the settings entered or contact their administrator.

SDP Start Phase

The start page is optional. Without the start page, during the SDP exchange, the user clicks the Next button on the welcome page and is sent to the registrar's introduction page. Because the user has not previously connected to the registrar, he or she is required to log in to the registrar using available credentials (per the registrar configuration). Some browsers fail to reconnect to the registrar after the user has entered the login data. As of Cisco IOS Release 12.4(4)T, users may configure their browsers to begin the SDP exchange by contacting the registrar's introduction URL through a start page. Thereafter, the registrar can direct the user to the welcome page, which is on the petitioner device. The SDP transaction continues through the welcome, introduction, and completion phases as described in this document.

To begin the SDP transaction from the registrar, the user must configure the browser through the **template http start** command; otherwise, the SDP transaction must begin from the welcome page on the petitioner. See the How Custom Templates Work with SDP, on page 1252.

Before the welcome page is displayed, the user must direct his or her browser to the start page through the URL http://registrar/ezsdd/intro. A login dialog box is then displayed, and the end user can log into the registrar through a username and password supplied by the administrator to access the secure network (see the figure below).

Figure 43: Registrar Remote Login Dialog Box

After entering a valid username and password, the start page is displayed (see the figure below).

Figure 44: Sample SDP Start Page

The user must log into the petitioner through the URL http://10.10.10.1/ezsdd/welcome. The welcome phase begins when the user clicks the Next button on the start page.

SDP Welcome Phase

The local login dialog box is then displayed (see the figure below), and the end user can log into the local device through the factory default username (cisco) and password (cisco). The welcome page is then displayed.

Figure 45: Petitioner Local Login Dialog Box

After the password is successfully entered, the welcome web page is displayed (see the figure below), which is served by the petitioner.

Figure 46: Sample SDP Welcome Page

After entering the URL of the registrar (for example, http://192.0.2.155/ezsdd/intro) and clicking the Next button on the welcome web page, the SDP introduction phase begins and the introduction page, which is served by the registrar, is displayed.

SDP Introduction Phase

Before the introduction page is displayed, the end user must log into the registrar if the user has not already done so from the start page (see "SDP Start Phase, on page 1245"), which utilizes the external AAA database.

With an external AAA database, the introducer can use an account on the database to perform the introduction without requiring knowledge of the enable password of the registrar. Without an external AAA database, the introducer may use the enable password of the registrar for authentication.

Note Using the enable password of the registrar exposes the password to end users; therefore, it is recommended that the enable password be used for administrative testing only.

The administrative introducer is identified by the HTTP authentication for the introduction page (or the start page), with the AAA database query returning administrative privilege for the user. If the introducer has

administrator privilege, the device name is that which was entered in the administrative introduction page. If the introducer does not have administrative privileges, the device name is the introducer name. The existing device certificate is the current certificate on the petitioner, which may be the manufacturing identification certificate (MIC). This certificate may or may not exist. For more information on the function of the external AAA database, see the section "How SDP Uses an External AAA Database, on page 1250."

After the end user successfully enters his or her password, the introduction web page is displayed (see the figure below).

Figure 47: Sample SDP Introduction Page

At this point, the registrar passes device information to the external management system to obtain a bootstrap configuration file. For more information on options available to identify a customized bootstrap configuration file, see the section Custom HTML Template Expansion Rules, on page 1253.

After the end user clicks the Next button on the introduction page, the end user enters the completion phase and automatically returns to his or her local device.

SDP Completion Phase

Now that the end user has enrolled the petitioner with the registrar, the petitioner serves the completion page (see the figure below).

Figure 48: Sample SDP Completion Page

The SDP exchange is now complete. The petitioner has received configuration information from the registrar and should receive a certificate from the registrar shortly.

SDP Leveraging USB Tokens

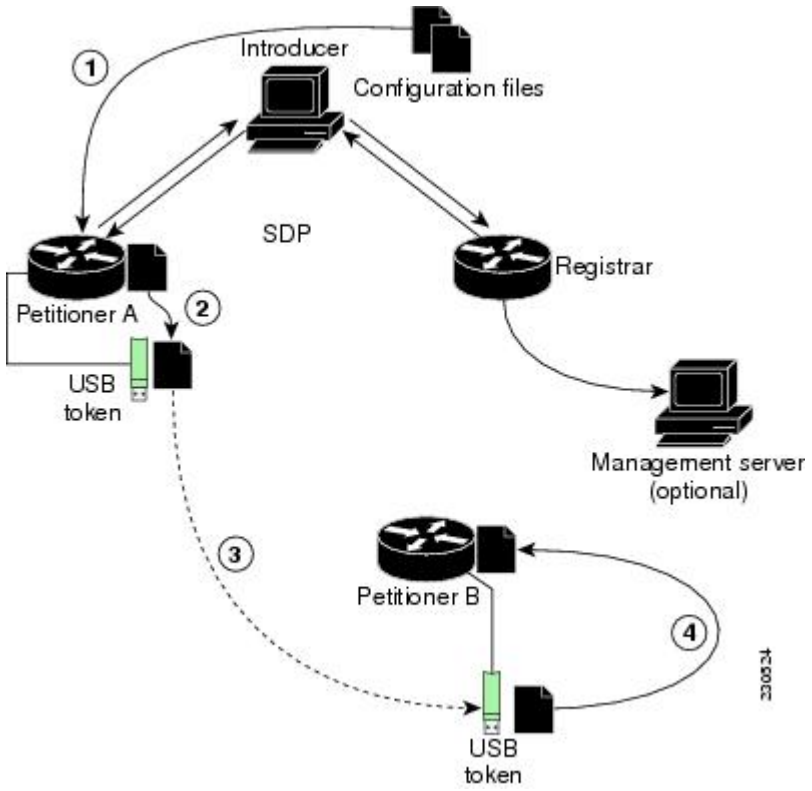
SDP provides for highly scalable deployments and streamlines the deployment of an individual device or multiple devices. USB tokens provide for secure storage and configuration distribution.

As of Cisco IOS Release 12.4(15)T or a later release, USB tokens may be utilized to transfer PKI credentials using SDP to a remote device, and SDP may be used to configure the USB token. The USB token may then be used to provision a device at the same location, or the USB token may be transported to another location where it may be used to provision a remote device.

An example SDP deployment using a USB token to transfer PKI credentials is shown in the figure below. The required devices include the USB token and the SDP entities required to provision a device. These SDP entities are the introducer, the registrar, a petitioner at the local location, Petitioner A, and a petitioner at the remote location, Petitioner B. Optionally, a management server may be used.

Note An optional configuration would be to configure one device as both the registrar and a petitioner, which may be beneficial when the USB token is transported to a remote location. The remote location would not require a separate petitioner device.

Figure 49: Example SDP Environment Using USB Tokens to Transfer Credentials



Use of SDP to Configure the USB Token

Prior to initiating an SDP introduction a USB token is inserted into the petitioner device. In the example configuration shown in the figure, the USB token would be inserted into Petitioner A. The petitioner may be configured to ignore any existing information on the USB token. As in regular SDP operations, for a scalable configuration of USB tokens, an initial template configuration has to be prepared and placed onto each SDP device with appropriate target configuration information.

Files used to provision a device are moved in the following sequence.

1. One petitioner, Petitioner A, is at the local location. petitioner A engages directly with the SDP exchange to perform the initial configuration of the USB token. Files used to configure the USB token, binary files and template files, are retrieved from the registrar and moved to Petitioner A.

The URL for the binary file location is expanded on the registrar. Binary files are not processed through the template expansion functions. The template expansion occurs on the registrar for both the source URL and destination URL.

By default, binary files and template files are retrieved from and stored to NVRAM on the registrar and petitioner respectively. The binary file location on the registrar and the destination binary file location on Petitioner A may be specified with the **binary file**command. The template file location on the registrar and the destination template file location on Petitioner A may be specified with the **template file**command.

1. The Rivest, Shamir, and Adelman (RSA) keys and certificate chain information are moved from Petitioner A to the USB token.
2. The USB token is transported to the remote location where it is inserted into Petitioner B.
3. The configuration files on the USB token are used to provision the local device. Files from the USB token may be moved to a storage location on Petitioner B with the **crypto key move rsa** command.

SDP Phases with a USB Token

The same SDP phase concepts introduced in the "SDP Overview" section are used, with the following distinctions in the SDP welcome phase, the SDP introduction phase, and the SDP completion phase.

SDP Welcome Phase with a USB Token

The SDP welcome phase begins as usual, when an introduction is initiated by connecting to the welcome user interface. If there is an existing certificate on the USB token, it is used for signing the SDP exchange. Instead of a local RSA key pair, a new RSA key pair on the token is used.

Note The RSA key pair generation may take a substantial length of time, anywhere from 5 to 10 minutes if the key is generated on the token. The length of time is dependent on hardware key generation routines available on the USB token. An informative web page is presented to the introducer, indicating that RSA key pair generation is occurring.

The new key pair generated by Petitioner A is added to the USB token without removing any existing RSA key pairs. SDP AV pairs indicate both that a token is being used and if there is any token secondary configuration information. If an optional management server is in use, the AV pair information is used to determine if any special configuration commands are needed.

SDP Introduction Phase with a USB Token

The SDP Introduction phase begins with AV pairs being transferred to the registrar. When the registrar detects USB token related AV pairs, the registrar, if previously configured, may prepare configuration information destined for the USB token. Currently configuration commands are sent as a specific configuration files that are subsequently merged with the running configuration.

The administrator can leverage normal SDP configuration commands to configure the USB token. USB token information that should be configured includes the certificate, the bootstrap configuration, and the PIN number configuration.

SDP Completion Phase with a USB Token

At the beginning of the completion phase, the introduction proceeds with AV pairs being transferred to the petitioner. The various files are stored in the specified file system locations and then the existing configuration file processing proceeds. This ordering allows the configuration to take advantage of the new files that have been transferred.

Use of the Configured USB Token

After the USB token is configured by Petitioner A, it is transported from its current location to the remote location, where the second petitioner, Petitioner B is located. The USB token is inserted into the target device, Petitioner B, which then inherits the USB token configuration and cryptographic material from the USB token. The end user at the remote location must have the PIN number on the USB token. The PIN number is either the default factory PIN or the PIN number the administrator configured during the introduction phase.

How SDP Uses an External AAA Database

The external AAA database is accessed twice during the SDP exchange. The first time the AAA database is accessed, the introducer is authenticated; that is, when the registrar receives an introduction request through the secure HTTP (HTTPS) server, the registrar does an AAA lookup based on the introducer's username and password to authorize the request. The second time the AAA database is accessed, authorization information is obtained and applied to the configuration and certificates that are issued to the petitioner device; that is, the registrar checks the integrity of the request by verifying the request signature using the petitioner-signing certificate. The certificate subject name may be specified in the AAA database, and up to nine configuration template variables may be specified and expanded into the template configuration.

Use of a Self-Signed Certificate Versus a Certificate Issued by Another CA Server

By default, the SDP exchange results in only one certificate being issued to the petitioner device. Although just one certificate is issued, the introducer is not restricted from introducing multiple devices and thus obtaining multiple certificates. By specifying the subject name in the certificate that is issued, you can be assured that all certificates that are issued in this way are associated with the introducer. You can use PKI AAA integration to further restrict the use of these certificates. Additionally, the AAA database can be configured to accept only one authentication and authorization request per user.

Because the petitioner certificate is self-signed, it is just used to convey the public key of the petitioner. No verification or authorization check is performed on the certificate; thus, authorization is per-user based and no per-device information is used.

There are some scenarios when per-device authorization is preferred. Therefore, if the petitioner is able to use certificates issued by other certification authority (CA) servers for SDP transactions, the existing PKI can be used and authorization can be achieved over the certificate attributes.

Configuring the petitioner and the registrar for certificate-based authorization provides authorization of the specific device being deployed. Previously, introducer-to-petitioner device communication was secured only using physical security between the introducer and the petitioner device. SDP certificate-based authorization gives the registrar an opportunity to validate the current device identity before accepting the introduction.

Authentication and Authorization Lists for SDP

When you are configuring your SDP registrar, if you specify an authentication list and an authorization list, the registrar uses the specified lists for all introducer requests. The authentication list is used when authenticating the introducer (the AAA server checks for a valid account by looking at the username and password). The

authorization list is used to receive the appropriate authorized fields for the certificate subject name and a list of template variables to be expanded into the Cisco IOS command-line interface (CLI) snippet that is sent back to the petitioner. The authentication and authorization lists are usually point to the same AAA server list, but it is possible to use a different database for authentication and authorization. (Storing files on different databases is not recommended.)

When a petitioner makes an introduction request, multiple queries are sent to the AAA list database on the RADIUS or TACACS+ server. The queries search for entries of the following form:

```
user Password <userpassword>
   cisco-avpair="ttti:subjectname=<<DN subjectname>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#=<<value>>"
```

Note The existence of a valid AAA username record is enough to pass the authentication check. The "cisco-avpair=tti" information is necessary only for the authorization check.

If a subject name was received in the authorization response, the SDP registrar stores it in the enrollment database, and that "subjectname" overrides the subject name that is supplied in the subsequent certificate request (PKCS10) from the petitioner device.

The numbered "tti:iosconfig" values are expanded into the SDP Cisco IOS snippet that is sent to the petitioner. The configurations replace any numbered ($1 through $9) template variable. Because the default Cisco IOS snippet template does not include the variables $1 through $9, these variables are ignored unless you configure an external Cisco IOS snippet template. To specify an external configuration, use the **template config** command.

Note The template configuration location may include a variable "$n," which is expanded to the name with which the user is logged in.

Authentication and Authorization Lists for an Administrative Introducer

The SDP mechanisms assume a permanent relationship between the introducer and the device. As a result, the introducer username is used to define the device name.

In some SDP deployment scenarios, the introducer is an administrator doing the introduction for many devices. However, using the introducer (the administrator) name to define the device name results in multiple devices being incorrectly deployed with the same device name. Instead, an administrative introducer allows the administrator to specify the correct device name during the introduction.

More generally stated, the introducer username is used as the database record locator to determine all other information about the device including the Cisco IOS configuration template, various template variables (pulled from an AAA database and expanded into the template), and the appropriate subject name for PKI certificates issued to the device. For simplicity, this database record locator is called the user/device name.

The administrative introducer provides a device name. In that way, an administrator can provide the appropriate record locator when doing an introduction. For example, if an administrator is trying to introduce a device for username "user1," the administrator introduces the device into the PKI network and provides user1 as the record locator after logging into the registrar using the administrator's own credentials. The record locator,

user1, becomes the device name. All other template and PKI certificate subject name information specific to the introduction is then provided by the user1 username records instead of by the administrator's record.

The registrar device uses the supplied username information with a user introducer name. The username allows the existing mechanisms for determining a user's authorization, template, and PKI certificate information to be supported without modification.

How Custom Templates Work with SDP

You may use custom templates to streamline the SDP process.

- Custom templates allow you to complete the web pages with the required start information, so the introducer is no longer required to contact the registrar and can immediately begin the SDP transaction.
- Custom templates allow customized deployment information to be displayed on the web pages, thereby tailoring the user experience.

An easy way to define a custom template is to modify the default template. Without custom templates, the introducer must contact the registrar for information to begin the SDP transaction. For a list of the default templates, see the sectionDefault Templates for SDP Transaction Web Pages, on page 1256.

Note It is recommended that only advanced SDP users configure custom templates because problems can result from modifying templates incorrectly before the templates are displayed in the introducer's browser.

Custom Template Variable Expansion

There are expansion variables in the templates that are replaced by the Cisco IOS SDP registrar or petitioner. These variables are expanded as follows:

- $$--"$"
- $a--attribute-value (AV) pairs
- $c--Trusted certificate
- $d--Dump AV pairs in browser
- $h--Hostname
- $k--Keylabel or "tti"
- $l--Trustpoint label = "tti"
- $n--HTTP client's username
- $s--Default TTI key size
- $t--Trustpoint configuration
- $u--Completion URL
- $1 to $9--Variables retrieved from AAA server during user authentication

Custom Template Variable Expansion Rules

Configuration and templates are used during an SDP exchange. Prior to use and after distribution, these templates are expanded using the following rules based in the SDP communication stage.

Custom HTML Template Expansion Rules

HTML templates are expanded immediately before being served to the HTTP client. The HTTP templates are expanded as follows:

- $u--Completion url, which is be populated with the SDP completion URL (for example: http://10.10.10.1/ezsdd/completion). This variable is used internally by SDP as the internal “wizard” state. It is expected that the SDP introduction page include something similar to the following text: “<FORM action=\“$u\”method=\“post\”>” for normal wizard processing.
- $n--introducer name or the device name entered by the administrative introducer.
- $$--$
- $h--Hostname
- $a--All AV pairs with or without a specified template character are written in the following HTML form format. (Because these AV pairs are not “INPUT type=hidden,” they are directly displayed on the web page for debugging templates or the SDP process.)

<INPUT type=hidden NAME=“attribute string here”

value=“variable string here”>

all HTML templates should have this!

$d = dump all av pairs in: attribute = value

URL Template Expansion Rules

There are URLs for the configuration template source, the file template source, and the file destination. These variables are expanded when the registrar prepares the URL, just before retrieving the configuration or file. For the file destination, these variables are expanded just before the petitioner copies the file to the file destination.

- $$--$
- $h--Hostname

URL Template Expansion Rules for iPhone Deployment

The following template expansion variables are introduced for iPhone deployment:

- $o - challenge password. This template character is expanded by the SDP registrar after it obtains the challenge password from the Simple Certificate Enrollment Protocol (SCEP) server, before the configuration profile is sent to the iPhone in the START phase.
- $i - unique device identifier (UDID) of the iPhone. This template character is expanded by the SDP registrar into the CN field of the Subject Name, before the configuration profile is sent to the iPhone in the INTRODUCTION phase.
- $p - subject name differentiator. This template character is expanded by the SDP registrar using the value configured through the CLI. Seethe Configuring the SDP Registrar to Deploy Apple iPhones, on page

1272 for more information. This value can be used to differentiate the two certificates issued by the SCEP server to the iPhone, one in the COMPLETION phase and one in the VPN establishment phase. You determine part and field of the Subject Name into which this value goes.

See the How SDP Deploys Apple iPhones in a PKI, on page 1259 for more information.

Custom Configuration and File Template Variable Expansion Rules

Custom configuration and file template variables are expanded both when the registrar prepares the configuration or file template and when the petitioner receives the configuration or file template.

Custom Configuration and File Template Variable Expansion Rules at the Registrar

When the registrar expands the configuration or file template, the following variables are used by the Cisco IOS CA. These variables are expanded before being sent through the SDP wizard.

- $$--$
- $h--Hostname
- $t--A simple default trustpoint configuration that includes $l, $k, and $s to be expanded at the client
- $1 to $9--Variables retrieved from AAA server during user authentication (not applicable to the file template)

Custom Configuration and File Template Variable Expansion Rules at the Petitioner

When the petitioner expands the configuration or file template, the following variables are expanded:

- $$--$
- $h--Hostname
- $k--Keylabel
- $l--Trustpoint label
- $s--Key size
- $c--Expanded to certificate chain
- $n--Expanded to username (not applicable to the file template)

Custom Configuration HTTP Template Variable Expansion Rules

Custom configuration HTTP templates provide flexibility for backend Common Gateway Interface (CGI) scripts and integration with external management systems. Template URLs run through the HTTP template expansions before registrar retrieves the bootstrap configuration from the external management system. The device name ($n) is expanded into the URL and passed to the external management system so that a specific bootstrap configuration file can be located based on the device information.

Note You should only modify the HTML text that is displayed. The existing expansion variables, Javascript, and forms in the default templates should not be removed when customizing the templates. They are required for SDP to function properly.

The HTTP template expansion and **template config** command allow you to specify either of the following file types to obtain a customized bootstrap configuration file:

- A configuration file based on the device name (for example, template config http://myserver/$n-config-file.conf)
- A CGI script based on the device name (for example, template config http://myserver/cgi-bin/mysdpcgi post)

As of Cisco IOS Release 12.4(6)T, the CGI support has been expanded so that the bootstrap configuration can be identified by not only the device name, but also the type, current Cisco IOS version information, and current configuration. This functionality expands the **template config** command with the **post** keyword, which tells the registrar to send this additional device information to the external management system through a CGI script with the HTTP or HTTPS protocol only.

The registrar passes the device information through AV pairs ($a) to the external management system. Using the AV pair information, the management system identifies the appropriate bootstrap configuration file and sends it back to the registrar. The additional AV pairs that are sent with the expanded CGI support for identification of the customized bootstrap configuration file are shown in the table below.

Table 150: AV Pairs Sent During HTTP Post to External Management System

AV Pair	Description
TTIFixSubjectName	AAA_AT_TTI_SUBJECTNAME (sent only if the realm authentication user is not the root user on the registrar)
TTIIosRunningConfig	Output of **show running-config brief**
TTIKeyHash	Digest calculated over the device public key
TTIPrivilege	AAA_AT_TTI_PRIVILEGE--”admin” is sent if the user is an administrator, “user” is sent if the user is not an administrator (sent only if the realm authentication user is an administrator and the information is available from the AAA server)
TTISignature	Digest calculated over all AV pairs except UserDeviceName and TTISignCert
TTISignCert	Device current certificate (sent only if the device currently has a certificate)
TTITemplateVar	AAA_AT_TTI_IOSCONFIG(1-9) (sent only if the realm authentication user is not the root user on the registrar)
TTIUserName	Device name
TTIVersion	TTI version of the registrar
UserDeviceName	Device name as entered by the administrative introducer (sent only if the realm authentication user is an administrator)

Note The registrar must be running Cisco IOS Release 12.4(6)T, the **template config** command must be issued with the **post** keyword, and the *url* argument must include either HTTP or HTTPS. No other protocol is supported for the expanded CGI template functionality (for example, FTP).

Default Templates for SDP Transaction Web Pages

The following default templates exist for each SDP transaction web page:

Default Prep-Connect Template

The prep-connect template may be modified by the administrator to contain values that are appropriate for their environment. The format of the prep-connect page may also be modified by the settings contained in the template.

Except for the registrar IP address, which the administrator must customize, the prep-connect template may be used as shown below.

```
<html><head><title>
SDP: Test Internet Connection</title></head>
<noscript><b>
If you see this message, your browser is not running JavaScript,<br>
which is required by Cisco Secure Device Provisioning.<br>
If you cannot enable JavaScript, please contact your system administrator.
<br><br></b></noscript>
<body style="background-color: rgb(204, 255, 255);">
<div style="text-align: center;"><big><big>
Secure Device Provisioning</big><br>
Test Internet Connection</big><br><br>
<form action="http://10.10.10.1/ezsdd/connect" method="post">
<input type="submit" value="Log onto Cisco Device"><br><br>
Default username/password is cisco/cisco.
<input type="hidden" name="TTIAfterConnectURL"
value="http://10.10.10.1/ezsdd/welcome">
<!-- Note, that for the below, 198.133.219.25 = www.cisco.com. -->
<input type="hidden" name="TTIConnectTestURL" value="http://198.133.219.25">
<input type="hidden" name="TTIInsideAddr" value="10.10.10.1">
<input type="hidden" name="TTIlanport" value="Vlan1">
<input type="hidden" name="TTIwanport" value="FastEthernet4">
</form></div></body></html>
```

Hidden HTML Form Fields

The hidden HTML form fields communicate initial configuration information to the browser as set by the administrator and are not signed.

Note The term “hidden” refers to the fact that these HTML form fields are not displayed on the prep-connect page to reduce potential confusion to the introducer.

The administrator can set hidden HTML form fields in the prep-connect template as shown in the table below.

Table 151: Administrator Defined AV Pairs Sent During Prep-Connect Phase

AV Pair	Description
TTIAfterConnectURL	The administrator may set the TTIAfterConnectURL field to either the welcome page URL or the start page URL. The welcome page URL is specified with the factory default petitioner IP address. The connect after URL may be any valid URL if SDP is not going to be used after establishing Internet connectivity.
TTIConnectTestURL	The administrator may set the TTIConnectTestURL field to a valid URL that should be accessible when Internet connectivity is established. The default prep-connect template value is www.cisco.com (198.133.219.25).
TTIInsideAddr	The administrator may set the TTIInsideAddr field to the factory default IP address of the petitioner. For the Cisco 871 ISR, the IP address is 10.10.10.1.
TTIlanportx	The administrator may set the TTIlanportx field to the LAN interface name of the petitioner platform. This field is used to apply the Cisco IOS connect configuration. For the Cisco 871, the field value is “Vlan1.”
TTIwanport	The administrator may set the TTIwanport field to the WAN interface name of the petitioner. This field is used to apply the Cisco IOS connect configuration. For the Cisco 871, the field value is “FastEthernet4.”

Note The connect template cannot be customized.

Default Start Page Template

```
<html><head><title>EZ-Secure Device Deployment Start page on $h</title></head>
<NOSCRIPT><B>
If you see this message, your browser is not running JavaScript.<BR>
Cisco Secure Device Deployment requires JavaScript.<BR> Please contact
your system adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function submit_to_url(form){
form.action=form.TTIWelcomeURL.value;return true;}</SCRIPT>
<B>Welcome to Cisco Secure Device Deployment Server $h</B> <FORM
action="" method="post" onSubmit="return submit_to_url(this)"> Your
device:<BR> <INPUT type="text" name="TTIWelcomeURL" size=80
value=\"\"><BR><BR> <INPUT type="submit" value="Next"><BR>
$a</FORM></html>
```

Default Welcome Page Template

```
<html><head><title>EZ-Secure Device Deployment WELCOME to $h</title></head>
<NOSCRIPT><B>
If you see this message, your browser is not running JavaScript.<BR>
Cisco Secure Device Deployment requires JavaScript.<BR> Please contact
your system adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE=\"JavaScript\">
function submit_to_url(form){
natURL=location.href.split(\"/\");
localURL=form.TTICompletionURL.value.split(\"/\");
if(natURL[2]!=localURL[2]){
```

```
form.TTICompletionURL.value=localURL[0]+\"//\"+natURL[2]+\"/
\"+localURL[3]+
\"/\"+localURL[4];}
form.action=form.vpnserviceurl.value;
return true;}</SCRIPT>
<B>Welcome to Cisco Secure Device Deployment for $h</B> <FORM
action=\"\" method=\"post\" onSubmit=\"return submit_to_url (this)\">
To join a Virtual Private Network (VPN) enter the web<BR> site URL
provided by your network administrator:<BR> <INPUT type=\"text\"
name=\"vpnserviceurl\" size=80 value=\"\"><BR><BR><INPUT
type=\"submit\" value=\"Next>\"><BR> $a</FORM></html>
```

Default Introduction Page Template

```
<html><head><title>EZ-Secure Device Deployment INTRODUCTION to $h</title>
</head><B>Welcome to the VPN network gateway on $h</B> <FORM
action=\"$u\" method=\"post\"> Your 'username' and 'password' entered
have been accepted.<BR> Your device will now be allowed to
automatically join the VPN network.<BR> <BR>Press Next to complete
automatic configuration of your VPN Device.<BR> <BR><INPUT
type=\"submit\" value=\"Next>\"><BR> $a</P></FORM></html>
```

Default Admin-Introduction Page Template

```
<html><head><title>EZ-Secure Device Deployment ADMINISTRATIVE
INTRODUCTION to $h</title></head> <NOSCRIPT><B> If you see this
message, your browser is not running JavaScript.<BR> Cisco Secure
Device Deployment requires JavaScript.<BR> Please contact your system
adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE=\"JavaScript\">
function submit_to_url(form){
form.introadminurl.value=location.href+\"/admin\";
form.action=form.introadminurl.value;
return true;}</SCRIPT>
<B>Welcome to the VPN network gateway on $h</B> <FORM action=\"\"
method=\"post\" onSubmit=\"return submit_to_url (this)\"> Your
administrator 'username' and 'password' entered have been
accepted.<BR> Please provide the name to be associated with this
device:<BR> <INPUT type=\"text\" name=\"userdevicename\" size=64
value=\"\"><BR><BR> <INPUT type=\"submit\" value=\"Next>\"><BR> <INPUT
type=\"hidden\" name=\"introadminurl\" value=\"\"><BR>
$a</FORM></html>
```

Default Completion Page Template

```
<html><head><title>EZ-Secure Device Deployment COMPLETE on $h</title></head>
<B>Now enrolling $h with the VPN network...</B><BR> Full network VPN
access should be available in a moment.<BR><BR> $d<BR></html>
```

Default Template for the Configuration File

The default configuration template is shown below. This default configuration file is used if a configuration template is not specified or if the **template config** command is issued without the **post** keyword. For more information on using the default configuration template, see the UsingaConfigurationTemplateFile Example, on page 1288.

```
$t
!
$c
```

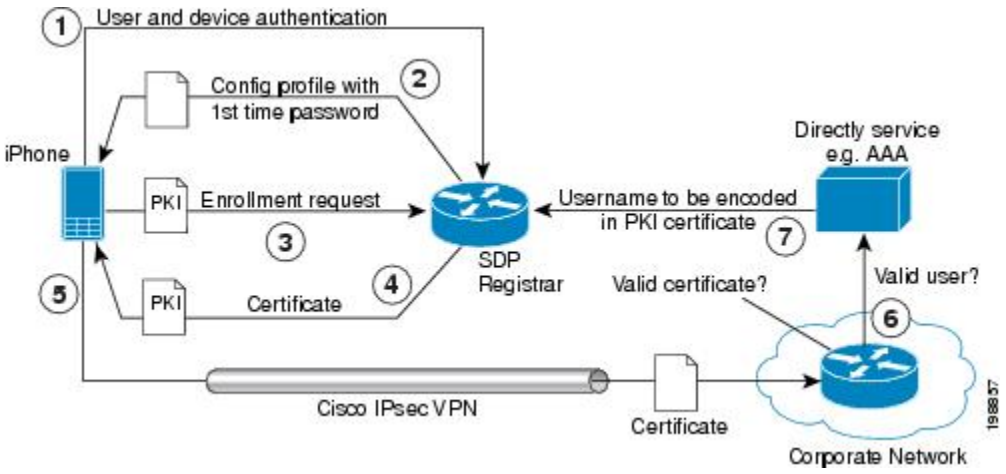
```
!
end
```

How SDP Deploys Apple iPhones in a PKI

With the introduction of the Cisco IOS 15.1(2)T and Apple iPhone OS 3.0 releases, Apple iPhones are supported on Cisco IOS network devices. Cisco IOS routers use the SDP registrar to deploy iPhones so that network applications can be accessed securely through an IPSec VPN, SCEP server, and PKI certificate deployment technologies.

The Apple iPhone combines the distribution of its XML-based "Configuration Profiles" with the initial deployment of certificates. SDP uses these initial certificates to authenticate access to enterprise applications and encrypt subsequent profile distribution. SDP uses this enrollment solution for distributing digital certificates to the iPhone.

Figure 50: SDP Registrar Deployment of the iPhone in a PKI



SDP Registrar Deployment Phases of the Apple iPhone in a PKI

The following sections describe each phase of the SDP registrar deployment of the iPhone in a PKI:

Start SDP Deployment Phase

The following steps describe the Start SDP deployment phase:

Note The Start SDP deployment phase is equivalent to the "Begin Enrollment" phase (or Phase 1) discussed in the http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf Apple iPhone Enterprise Deployment Guide .

SUMMARY STEPS

1. The iPhone user opens the Safari browser and types the start page HTTPS URL. For example, this HTTPS URL may be an internal corporate network address. The SDP registrar HTTPS page initiates the process.
2. The user starts authentication with the Cisco router, which acts as the SDP registrar by providing a username and password.

3. The SDP registrar contacts the SCEP server to obtain a challenge password.
4. The SDP registrar constructs a configuration profile in XML format that consists of the challenge password, SCEP server URL, and a request for iPhone attributes. The SCEP server URL is used to send the enrollment request and the iPhone device attributes are used by the iPhone to generate the RSA keys.
5. The iPhone user installs the configuration profile on the iPhone to complete the Start SDP phase.

DETAILED STEPS

Step 1 The iPhone user opens the Safari browser and types the start page HTTPS URL. For example, this HTTPS URL may be an internal corporate network address. The SDP registrar HTTPS page initiates the process.

Step 2 The user starts authentication with the Cisco router, which acts as the SDP registrar by providing a username and password.

Step 3 The SDP registrar contacts the SCEP server to obtain a challenge password.

Step 4 The SDP registrar constructs a configuration profile in XML format that consists of the challenge password, SCEP server URL, and a request for iPhone attributes. The SCEP server URL is used to send the enrollment request and the iPhone device attributes are used by the iPhone to generate the RSA keys.

The following example shows a configuration profile sent by the SDP registrar to the iPhone in the Start SDP deployment phase:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>PayloadContent</key>
<dict>
<key>URL</key>
<string>https://profileserver.example.com/iphone</string>
<key>DeviceAttributes</key>
<array>
<string>UDID</string>
<string>IMEI</string>
<string>ICCID</string>
<string>VERSION</string>
<string>PRODUCT</string>
</array>
<key>Challenge</key>
<string>optional challenge</string>
```

Step 5 The iPhone user installs the configuration profile on the iPhone to complete the Start SDP phase.

Welcome SDP Deployment Phase

The Welcome SDP deployment phase is not applicable for the iPhone because the Introducer (for example, Safari web browser) is run on the SDP petitioner (iPhone).

Introduction SDP Deployment Phase

The following steps describe the Introduction SDP deployment phase:

Note The Introduction SDP deployment phase is equivalent to the “Device Authentication” phase .

SUMMARY STEPS

1. The iPhone triggers an HTTPS post containing the requested device attribute information and the challenge password as a configuration profile. The HTTPS post is directed to the HTTPS URL specified in the configuration profile obtained in the Start SDP deployment phase, which must be the Introduction SDP deployment phase URL. The post data is signed by the iPhone using an Apple-issued certificate (built-in identity) and this signature may be verified, the identify confirmed, and the device attributes checked.
2. The UDID sent by the iPhone is captured by the SDP registrar and included in the Subject Name. Going forward, the device attributes obtained by the SDP registrar are used to determine if this was exactly the type of device that would be accepted. For example, the network administrator would only let 3GS iPhones onto the network because they have hardware encrypted storage. The device attributes obtained would enable the SDP registrar to distinguish 3GS iPhones from 3G iPhones.
3. The SDP registrar responds by building a configuration profile that consists of the following: HTTP URL of the SCEP server, Subject Name (contains the UDID) that is sent in the enrollment request, key size, key type, key usage, and challenge password. If the START phase had been skipped, the SDP registrar would contact the SCEP server to obtain a challenge password. See the URL Template Expansion Rules for iPhone Deployment, on page 1253 for more information about how the SDP registrar obtains the Subject Name and the challenge password.

DETAILED STEPS

Step 1 The iPhone triggers an HTTPS post containing the requested device attribute information and the challenge password as a configuration profile. The HTTPS post is directed to the HTTPS URL specified in the configuration profile obtained in the Start SDP deployment phase, which must be the Introduction SDP deployment phase URL. The post data is signed by the iPhone using an Apple-issued certificate (built-in identity) and this signature may be verified, the identify confirmed, and the device attributes checked.

Step 2 The UDID sent by the iPhone is captured by the SDP registrar and included in the Subject Name. Going forward, the device attributes obtained by the SDP registrar are used to determine if this was exactly the type of device that would be accepted. For example, the network administrator would only let 3GS iPhones onto the network because they have hardware encrypted storage. The device attributes obtained would enable the SDP registrar to distinguish 3GS iPhones from 3G iPhones.

The following example shows a configuration profile sent by the iPhone in the Introduction SDP deployment phase:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
  DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>UDID</key>
<string></string>
<key>VERSION</key>
<string>7A182</string>
<key>MAC_ADDRESS_EN0</key>
<string>00:00:00:00:00:00</string>
<key>CHALLENGE</key>
```

```
either:
    <string>String</string>
or:
    <data>"base64 encoded data"</data>
</dict>
</plist>
```

Step 3 The SDP registrar responds by building a configuration profile that consists of the following: HTTP URL of the SCEP server, Subject Name (contains the UDID) that is sent in the enrollment request, key size, key type, key usage, and challenge password. If the START phase had been skipped, the SDP registrar would contact the SCEP server to obtain a challenge password. See the URL Template Expansion Rules for iPhone Deployment, on page 1253 for more information about how the SDP registrar obtains the Subject Name and the challenge password.

Note The SDP registrar supports the RSA key type only.

The following example shows a configuration profile sent by the SDP registrar in the Introduction SDP deployment phase:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>PayloadContent</key>
<dict>
<key>URL</key>
<string>https://iphone.vpn.apple.com/pkifoobar.exe</string>
<key>Name</key>
<string>instance_for_getcacert_call</string>
<key>Subject</key>
<array>
<array>
<array>
<string>O</string>
<string>Apple Inc.</string>
</array>
</array>
<array>
<array>
<string>CN</string>
<string>Foo</string>
</array>
</array>
</array>
<key>Challenge</key>
<string>CHALLENGE</string>
<key>Keysize</key>
<integer>1024</integer>
<key>Key Type</key>
<string>RSA</string>
<key>Key Usage</key>
<integer>5</integer>
</dict>
<key>PayloadDescription</key>
<string>Provides device encryption identity</string>
<key>PayloadUUID</key>
<string>fd8a6b9e-0fed-406f-9571-8ec98722b713</string>
<key>PayloadType</key>
<string>com.apple.security.scep</string>
```

```
<key>PayloadDisplayName</key>
<string>Encryption Identity</string>
<key>PayloadVersion</key>
<integer>1</integer>
<key>PayloadOrganization</key>
<string>Apple Inc.</string>
<key>PayloadIdentifier</key>
<string>com.apple.encrypted-profile-service</string>
</dict>
</plist>
```

Post-Introduction SDP Deployment Phase

The following steps describe the Post-introduction SDP deployment phase.

Note The Post-introduction SDP deployment phase is equivalent to the "Certificate Installation" phase (or Phase 3) discussed in the http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf Apple iPhone Enterprise Deployment guide .

SUMMARY STEPS

1. The iPhone installs the configuration profile specification containing SCEP information obtained from the SDP registrar in the Introduction SDP deployment phase.
2. The iPhone generates the keys with the instructions in the profile specification and sends the enrollment request to the SCEP server whose HTTP URL is specified in the profile, along with the challenge password.
3. The SCEP server verifies the challenge password and issues the digital certificate to the iPhone.
4. The user can install this certificate on the iPhone and use the Cisco IPsec VPN to connect to the corporate network.

DETAILED STEPS

Step 1 The iPhone installs the configuration profile specification containing SCEP information obtained from the SDP registrar in the Introduction SDP deployment phase.

Step 2 The iPhone generates the keys with the instructions in the profile specification and sends the enrollment request to the SCEP server whose HTTP URL is specified in the profile, along with the challenge password.

Step 3 The SCEP server verifies the challenge password and issues the digital certificate to the iPhone.

Step 4 The user can install this certificate on the iPhone and use the Cisco IPsec VPN to connect to the corporate network.

Note This certificate can also be used to download other enterprise settings, such as VPN settings, and Wi-Fi settings.

Second-Introduction SDP Deployment Phase

The following steps describe the Second-introduction SDP deployment phase:

Note The Second-introduction SDP deployment phase is equivalent to the "Device Configuration" phase (or Phase 4) discussed in the http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf Apple iPhone Enterprise Deployment guide .

SUMMARY STEPS

1. The iPhone repeats the Introduction SDP deployment phase with the following exceptions:
2. The SDP registrar responds with a configuration profile that includes the general enterprise settings such as VPN settings, Wi-Fi settings, and email settings. and in addition includes SCEP settings for a second certificate to be used for establishing a VPN.

DETAILED STEPS

Step 1 The iPhone repeats the Introduction SDP deployment phase with the following exceptions:

- The iPhone does not include the challenge password as part of the post data .
- The iPhone signs the post data using the certificate obtained from the SCEP server in the Post-introduction SDP deployment phase.

Step 2 The SDP registrar responds with a configuration profile that includes the general enterprise settings such as VPN settings, Wi-Fi settings, and email settings. and in addition includes SCEP settings for a second certificate to be used for establishing a VPN.

Second Post-Introduction SDP Deployment Phase

The Second Post-introduction SDP phase is identical to the Post-introduction SDP deployment phase. The iPhone generates a certificate request based on the SCEP settings provided by the SDP registrar in the Second-introduction SDP deployment phase and enrolls with the SCEP server.

Completion SDP Deployment Phase

The Completion SDP deployment phase is not applicable for the iPhone because the Introducer (for example, the Safari web browser) is run on the SDP petitioner (iPhone).

How to Set Up Secure Device Provisioning (SDP) for Enrollment in a PKI

This section contains the following procedures that should be followed when setting up SDP for your PKI. You can configure the registrar according to only one of the registrar configuration tasks.

Enabling the SDP Petitioner

Perform this task to enable or disable the petitioner and associate a trustpoint with the SDP exchange.

You can also use this task to configure the petitioner to use a certificate and the RSA keys associated with a specific trustpoint.

Note The petitioner is enabled by default on a Cisco device that contains a crypto image; thus, you have only to issue the **crypto provisioning petitioner** command if you have previously disabled the petitioner or if you want to use an existing trustpoint instead of the automatically generated trustpoint.

Note By default, the SDP petitioner device uses an existing certificate. If multiple certificates and one specific certificate exist, use this task to make a choice. However, this task is not necessary to enable the default behavior.

Before you begin

- The HTTP server must be enabled through the **ip http server**command. (The HTTP server is typically enabled by default in many default Cisco IOS configurations.)
- If you are configuring the petitioner to use a certificate and RSA keys, your SDP petitioner device must have an existing manufacturer's certificate or a third-party certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning petitioner**
4. Do one of the following:
   - **trustpoint** *trustpoint-label*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning petitioner** **Example:**	Allows SDP petitioner device behavior to be modified and enters tti-petitioner configuration mode.

	Command or Action	Purpose
	`Router(config)# crypto provisioning petitioner`	**Note** Effective with Cisco IOS Release 12.3(14)T, the **crypto provisioning petitioner**command replaced the **crypto wui tti petitioner**command.

Step 4	Do one of the following:
	• **trustpoint** *trustpoint-label*
	Example:
	`Router(tti-petitioner)# trustpoint mytrust`
	Example:
	Example:
	Example:
	trustpoint signing
	trustpoint-label
	Example:
	`Router(tti-petitioner)# trustpoint signing mytrust`
		(Optional) Specifies the trustpoint that is to be associated with the SDP exchange between the petitioner and the registrar.
		Note If this command is not issued, the *trustpoint-label*argument is aut omatically labeled "tti."
		(Optional) Specifies the trustpoint and associated certificate that are used when signing all introduction data during the SDP exchange.

Step 5	**end** **Example:** `Router(tti-petitioner)# end`	(Optional) Exits tti-petitioner configuration mode.

Troubleshooting Tips

After the SDP exchange is complete, a new trustpoint-label named "tti" exists. The trustpoint is automatically enrolled with the certificate server (the registrar). To verify that the trustpoint is really there, use the **show running-config** command.

What to Do Next

If you set up the petitioner to use a certificate and the RSA keys associated with the specified trustpoint, you should configure the registrar as shown in the task "Enabling the SDP Registrar for Certificate-Based Authorization."

Enabling the SDP Registrar and Adding AAA Lists to the Server

Perform this task to enable the registrar and associate a certificate server with the SDP exchange.

You can also use this task if you want to add an authentication list and an authorization list to the RADIUS or TACACS+ server.

Prerequisites

Before configuring a registrar, perform the following tasks:

- Enable the HTTP server or the HTTPS server.

Note Before you enable an HTTPS server, you must disable the standard HTTP server if it is configured. Use the the **no ip http server** command to disable an HTTP server. To enable an HTTPS server, you should issue the **ip http secure-server** command followed by the **ip http secure-trustpoint** command. The specified trustpoint is a registrar local trustpoint appropriate for HTTPS communication between the registrar and the user's browser.

- Configure the Cisco IOS certificate server through the **crypto pki server** command.

If you are configuring AAA lists, you should complete the prerequisites required for the registrar in addition to completing the following tasks:

- Add user information to the AAA server database. To configure a RADIUS or TACACS+ AAA server, see the "Configuring RADIUS" and "Configuring TACACS+ " chapters of t he *Cisco IOS Security Configuration Guide* .
- Configure new AAA lists. To configure AAA lists, see the following chapters in the *Cisco IOS Security Configuration Guide* : "Configuring RADIUS," "Configuring TACACS+," "Configuring Authentication," and "Configuring Authorization ."

Restrictions

Cisco IOS CA Device Requirement

During the SDP process, a Cisco IOS CA certificate is automatically issued to the peer device. If an SDP registrar is configured on a third-party vendor's CA device, the SDP process does not work.

The template config Command

There are nine Cisco IOS configuration variables. If you require more configuration flexibility, the **template config** command can be used to reference a configuration template that is specific to the introducer. For more information on configuration flexibility, see the "Custom Configuration and File Template Variable Expansion Rules, on page 1254" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto provisioning registrar**
4. **pki-server** *label*
5. **authentication list** *list-name*
6. **authorization list** *list-name*
7. **template username** *name* **password** *password*
8. **template config** *url* [**post**]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. **Note** Effective with Cisco IOS Release 12.3(14)T, the **crypto provisioning registrar**command replaced the **crypto wui tti registrar**command.
Step 4	**pki-server** *label* **Example:** `Router(tti-registrar)# pki-server mycs`	Specifies the certificate server that is to be associated with the SDP exchange between the petitioner and the registrar.
Step 5	**authentication list** *list-name* **Example:** `Router (tti-registrar)# authentication list authen-tac`	(Optional) Authenticates the introducer in an SDP exchange.
Step 6	**authorization list** *list-name* **Example:** `Router (tti-registrar)# authorization list author-rad`	(Optional) Receives the appropriate authorized fields for the certificate subject name and list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.
Step 7	**template username** *name* **password** *password* **Example:**	(Optional) Establishes a username and password in which to access the configuration template on the file system.

	Command or Action	Purpose
	`Router(tti-registrar)# template username ftpuser password ftppwd`	
Step 8	**template config** *url* [**post**] **Example:** `Router(tti-registrar)# template config http://myserver/cgi-bin/mycgi post`	(Optional) Specifies a remote URL for the Cisco IOS CLI configuration template. The *url* argument can reference a configuration file that allows you to specify the device name ($n) to identify a bootstrap configuration. CGI support allows you to reference a CGI script through either HTTP or HTTPS and identify the bootstrap configuration by not only the device name, but also by the type, current Cisco IOS version and current configuration. The **post** keyword must be used for CGI support. **Note** The registrar must be running Cisco IOS Release 12.4(6)T or later to utilize expanded CGI support. If the registrar is running an earlier version of Cisco IOS, the additional device identification information is ignored.
Step 9	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Examples

To help troubleshoot the SDP transaction, you can issue the **debug crypto provisioning**command, which displays output from the petitioner and registrar devices.

The following is output for the **debug crypto provisioning**command. The output from the petitioner and registrar devices are shown below.

```
Petitioner device
! The user starts the Welcome phase.
Nov  7 03:15:48.171: CRYPTO_PROVISIONING: received welcome get request.
! The router generates a Rivest, Shamir, and Adelman (RSA) keypair for future enrollment.
Nov  7 03:15:48.279: CRYPTO_PROVISIONING: keyhash 'A506BE3B83C6F4B4A6EFCEB3D584AACA'
! The TTI transaction is completed.
Nov  7 03:16:10.607: CRYPTO_PROVISIONING: received completion post request.
Registrar device
!. During the introduction phase, the browser prompts for login information.
06:39:18: CRYPTO_PROVISIONING: received introduction post request.
06:39:18: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist, ttiuser)
! This happens if the user types in the wrong username or password.
06:39:19: CRYPTO_PROVISIONING: authentication declined by AAA, or AAA server not found -
0x3
06:39:19: CRYPTO_PROVISIONING: aaa query fails!
! The user re-enters login information.
06:39:19: CRYPTO_PROVISIONING: received introduction post request.
06:39:19: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist, ttiuser)
```

```
06:39:20: CRYPTO_PROVISIONING: checking AAA authorization (ipsecca_script_aaalist, ttiuser)
! The login attempt succeeds and authorization information is retrieved from the AAA database.
06:39:21: CRYPTO_PROVISIONING: aaa query ok!
! These attributes are inserted into the configuration template.
06:39:21: CRYPTO_PROVISIONING: building TTI av pairs from AAA attributes
06:39:21: CRYPTO_PROVISIONING: "subjectname" = "CN=user1, O=company, C=US"
06:39:21: CRYPTO_PROVISIONING: "$1" = "ntp server 10.3.0.1"
06:39:21: CRYPTO_PROVISIONING: "$2" = "hostname user1-vpn"
! The registrar stores this subject name and overrides the subject name in the subsequent
enrollment request.
06:39:21: CRYPTO_PROVISIONING: subjectname=CN=user1, O=company, C=US
! The registrar stores this key information so that it may be used to automatically grant
the subsequent enrollment request.
06:39:21: CRYPTO_PROVISIONING: key_hash=A506BE3B83C6F4B4A6EFCEB3D584AACA
```

Enabling the SDP Registrar for Certificate-Based Authorization

Perform this task to enable the SDP registrar to verify the petitioner-signing certificate using either a specified trustpoint or any configured trustpoint and initiate authorization lookups using the introducer username and the certificate name field.

Before you begin

You must also configure the SDP petitioner to use a certificate and RSA keys associated with a specific trustpoint. To complete this task, use the trustpoint signing command as shown in the task "Enabling the SDP Petitioner, on page 1264."

Note Because RADIUS does not differentiate between authentication and authorization, you need to use the default password, cisco, for certificate authorization.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **template file** *sourceURL destinationURL*
5. **binary file** *sourceURL destinationURL*
6. **authentication trustpoint** {**trustpoint-label| use-any }**
7. **authorization {login | certificate | login certificate}**
8. **authorization username subjectname** *subjectname*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.
Step 4	**template file** *sourceURL destinationURL* **Example:** `Router(tti-registrar)# template file http://myserver/registrar_file_r1 http://myserver/petitioner_file_p1`	(Optional) Specifies the source template file location on the registrar and the destination template file location on the petitioner. **Note** This command is useful when using a USB token to provision a device. The template expansion occurs on the registrar for both the source URL and file content. The destination URL is expanded on the petitioner.
Step 5	**binary file** *sourceURL destinationURL* **Example:** `Router(tti-registrar)# binary file http://myserver/registrar_file_a1 http://myserver/petitioner_file_b1`	(Optional) Specifies the binary file location on the registrar and the destination binary file location on the petitioner. **Note** This command is useful when using a USB token to provision a device. Both the source and destination URL are expanded on the registrar. Also, the destination URL and file content are expanded on the petitioner. Binary files are not processed through the template expansion functions.
Step 6	**authentication trustpoint** {**trustpoint-label**| **use-any** } **Example:** `Router(tti-registrar)# authentication trustpoint mytrust`	(Optional) Specifies the trustpoint used to authenticate the SDP petitioner device's existing certificate. • *trustpoint-label* --Specifies a specific trustpoint. • **use-any** --Specifies any configured trustpoint. **Note** If you do not use this command to specify a trustpoint, the existing petitioner certificate is not validated. (This functionality provides compatibility with self-signed petitioner certificates.)
Step 7	**authorization {login | certificate | login certificate}** **Example:**	(Optional) Enables AAA authorization for an introducer or a certificate.

	Command or Action	Purpose
	`Router(tti-registrar)# authorization login certificate`	• Use the **login** keyword for authorization based on the introducer's username. • Use the **certificate** keyword for authorization based on the petitioner's certificate. • Use the **login certificate** keyword for authorization based on the introducer's username and the petitioner's certificate.
Step 8	**authorization username subjectname** *subjectname* **Example:** `Router(tti-registrar)# authorization username subjectname all`	Sets parameters for the different certificate fields that are used to build the AAA username. • The **all** keyword specifies that the entire subject name if the certificate is used as the authorization username.
Step 9	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Configuring the SDP Registrar to Deploy Apple iPhones

Perform this task to configure the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network.

Before you begin

Ensure that the SDP Registrar is enabled to run HTTPS. See the Enabling the SDP Registrar and Adding AAA Lists to the Server section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http secure-server**
4. **crypto provisioning registrar**
5. **url-profile start** *profile-name*
6. **url-profile intro** *profile-name*
7. **match url** *url*
8. **match authentication trustpoint** *trustpoint-name*
9. **match certificate** *certificate-map*
10. **mime-type** *mime-type*
11. **template location** *location*
12. **template variable p** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http secure-server** **Example:** `Router(config)# ip http secure-server`	Enables the HTTPS web server.
Step 4	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. **Note** Effective with Cisco IOS Release 12.3(14)T, the **crypto provisioning registrar**command replaced the **crypto wui tti registrar**command.
Step 5	**url-profile start** *profile-name* **Example:** `Router(tti-registrar)# url-profile start START`	Specifies the **start** keyword to indicate that a URL profile is to be associated with the Start SDP deployment phase. The *profile-name* argument specifies the name of a unique URL profile. **Note** Both the Introduction SDP deployment phase and the Start SDP deployment phase can use different profiles or use the same URL profile.
Step 6	**url-profile intro** *profile-name* **Example:** `Router(tti-registrar)# url-profile intro INTRO`	Specifies the **intro** keyword to indicate that a URL profile is to be associated with the Introduction SDP deployment phase. The *profile-name* argument specifies the name of a unique URL profile. **Note** Both the Introduction SDP deployment phase and the Start SDP deployment phase can use different profiles or use the same URL profile.
Step 7	**match url** *url* **Example:** `Router(tti-registrar)# match url /sdp/intro`	Specifies the URL to be associated with the URL profile.

	Command or Action	Purpose
Step 8	**match authentication trustpoint** *trustpoint-name* **Example:** `Router(tti-registrar)# match authentication trustpoint apple-tp`	(Optional) Specifies the trustpoint name that should be used to authenticate the peer's certificate. If the trustpoint name is not specified, then the trustpoint configured using the **authentication trustpoint command**in tti-registrar configuration mode is used to authenticate the peer's certificate. See the Enabling the SDP Registrar for Certificate-Based Authorization section for more information.
Step 9	**match certificate** *certificate-map* **Example:** `Router(tti-registrar)# match certificate cat 10`	(Optional) Specifies the name of the certificate map used to authorize the peer's certificate.
Step 10	**mime-type** *mime-type* **Example:** `Router(tti-registrar)# mime-type application/x-apple-aspen-config`	Specifies the Multipurpose Internet Mail Extensions (MIME) type that the SDP registrar should use to respond to a request received through this URL profile.
Step 11	**template location** *location* **Example:** `Router(tti-registrar)# template location flash:intro.mobileconfig`	Specifies the location of the template that the SDP Registrar should use while responding to a request received through this URL profile.
Step 12	**template variable p** *value* **Example:** `Router(tti-registrar)# template variable p iphone-vpn`	(Optional) Specifies the value that goes into the Organizational Unit (OU) field of the subject name in the trustpoint certificate to be issued by the SDP Registrar. See this field in the certificate presented in the Apple CA Server Trustpoint Certificate Configuration Example section below.

Apple CA Server Trustpoint Certificate Configuration

The SDP Registrar must verify the signature generated from the iPhone's trustpoint certificate in order to trust the Apple CA server certificate. The iPhone signs its messages using the trustpoint certificate, which is issued by Apple's CA server during the Introduction SDP deployment phase.

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method of the Apple CA certificate:

Note See also the "How to Configure Certificate Enrollment for a PKI" section in the Configuring Certificate Enrollment for a PKI feature module for more detailed information about configuring a trustpoint certificate.

SUMMARY STEPS

1. The**crypto pki trustpoint** command is entered in global configuration mode to declare the trustpoint and a given name and enters ca-trustpoint configuration mode:
2. The **enrollment terminal** command is entered to specify manual cut-and-paste certificate enrollment
3. The **crypto pki authenticate** command retrieves the CA certificate and authenticates it from the specified TFTP server.
4. Copy the following block of text containing the base 64 encoded Apple CA trust certificate and paste it at the prompt.
5. The **exit** command is used to exit ca-trustpoint configuration mode and enter global configuration mode.
6. The **crypto provisioning registrar** command is entered in global configuration mode to specify the router to become a registrar for the SDP exchange and enters tti-registrar configuration mode.
7. The **url-profile command with the intro** keyword is entered in tti-registrar configuration mode to specify the unique URL profile name that is associated with the Introduction SDP deployment phase.
8. The **match authentication trustpoint**command is entered in tti-registrar configuration mode to specify the trustpoint name that should be used to authenticate the peer's certificate.

DETAILED STEPS

Step 1 The**crypto pki trustpoint** command is entered in global configuration mode to declare the trustpoint and a given name and enters ca-trustpoint configuration mode:

Example:

```
Router(config)# crypto pki trustpoint apple-tp
```

Step 2 The **enrollment terminal** command is entered to specify manual cut-and-paste certificate enrollment

Example:

```
Router(ca-trustpoint)# enrollment terminal
```

Step 3 The **crypto pki authenticate** command retrieves the CA certificate and authenticates it from the specified TFTP server.

Example:

```
Router(ca-trustpoint)# crypto pki authenticate apple-tp
```

Step 4 Copy the following block of text containing the base 64 encoded Apple CA trust certificate and paste it at the prompt.

Example:

```
I Bag Attributes
    localKeyID: 7C 29 15 15 12 C9 CF F6 15 2B 5B 25 70 3D A7 9A 98 14 36 06
subject=/C=US/O=Apple Inc./OU=Apple iPhone/CN=Apple iPhone Device CA
issuer=/C=US/O=Apple Inc./OU=Apple Certification Authority/CN=Apple iPhone Certification Authority
-----BEGIN CERTIFICATE-----
MIIDaTCCAlGgAwIBAgIBATANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzET
MBEGA1UEChMKQXBwbGUgSW5jLjEmMCQGA1UECxMdQXBwbGUgQ2VydGlmaWNhdGlv
biBBdXRob3JpdHkxLTArBgNVBAMTJEFwcGxlIGlQaG9uZSBDZXJ0aWZpY2F0aW9u
IEF1dGhvcml0eTAeFw0wNzA0MTYyMjU0NDZaFw0xNDA0MTYyMjU0NDZaMFoxCzAJ
BgNVBAYTAlVTMRMwEQYDVQQKEwpBcHBsZSBJbmMuMRUwEwYDVQQLEwxBcHBsZSBp
UGhvbmUxHzAdBgNVBAMTFkFwcGxlIGlQaG9uZSBEZXZpY2UgQ0EwgZ8wDQYJKoZI
hvcNAQEBBQADgY0AMIGJAoGBAPGUSsnquloYYK3Lok1NTlQZaRdZB2bLl+hmmkdf
Rq5nerVKc1SxywT2vTa4DFU4ioSDMVJl+TPhl3ecK0wmsCU/6TKqewh0lOzBSzgd
```

```
Z04IUpRai1mjXNeT9KD+VYW7TEaXXm6yd0UvZ1y8Cxi/WblshvcqdXbSGXH0KWO5
JQuvAgMBAAGjgZ4wgZswDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8w
HQYDVR0OBBYEFLL+ISNEhpVqedWBJo5zENinTI50MB8GA1UdIwQYMBaAFOc0Ki4i
3jlga7SUzneDYS8xoHw1MDgGA1UdHwQxMC8wLaAroCmGJ2h0dHA6Ly93d3cuYXBw
bGUuY29tL2FwcGxlY2EvaXBob251LmNybDANBgkqhkiG9w0BAQUFAAOCAQEAd13P
Z3pMViukVHe9WUg8Hum+0I/0kHKvjhwVd/IMwGlXyU7DhUYWdja2X/zqj7W24Aq5
7dEKm3fqqxK5XCFVGY5HI0cRsdENyTP7lxSiiTRYj2mlPedheCn+k6T5y0U4Xr40
FXwWb2nWqCF1AgIudhgvVbxlvqcxUm8Zz7yDeJ0JFovXQhyO5fLUHRLCQFssAbf8
B4i8rYYsBUhYTspVJcxVpIIltkYpdIRSIARA49HNvKK4hzjzMS/OhKQpVKw+OCEZ
xptCVeN2pjbdt9uzi175oVo/u6B2ArKAW17u6XEHIdDMOe7cb33peVI6TD15W4MI
pyQPbp8orlXe+tA8JA==
-----END CERTIFICATE-----
```

Step 5 The **exit** command is used to exit ca-trustpoint configuration mode and enter global configuration mode.

Example:

```
Router(ca-trustpoint)# exit
```

Step 6 The **crypto provisioning registrar** command is entered in global configuration mode to specify the router to become a registrar for the SDP exchange and enters tti-registrar configuration mode.

Example:

```
Router(config)# crypto provisioning registrar
```

Step 7 The **url-profile command with the intro** keyword is entered in tti-registrar configuration mode to specify the unique URL profile name that is associated with the Introduction SDP deployment phase.

Example:

```
Router(tti-registrar)# url-profile intro INTRO
```

Step 8 The **match authentication trustpoint**command is entered in tti-registrar configuration mode to specify the trustpoint name that should be used to authenticate the peer's certificate.

Example:

```
Router(tti-registrar)# match authentication trustpoint apple-tp
```

The SDP Registrar can now use the Apple CA trustpoint certificate called "apple-tp" for verifying the signature of the iphone.

Configuring an Administrative Introducer

Perform the following task to configure an administrative introducer using administrator authentication and authorization lists.

Before you begin

The administrative introducer must have enable privileges on the client device and administrator privileges on the server.

Note When using RADIUS, a user/device that needs to be introduced by the administrative introducer must always use cisco as its own password. TACACS+ does not have this limitation; a user/device can have any password and be introduced by the administrative introducer.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **administrator authentication list** *list-name*
5. **administrator authorization list** *list-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.
Step 4	**administrator authentication list** *list-name* **Example:** `Router(tti-registrar)# administrator authentication list authen-tac`	Configures the AAA list used to authenticate an administrator during an introduction.
Step 5	**administrator authorization list** *list-name* **Example:** `Router(tti-registrar)# administrator authorization list author-tac`	Configures the AAA list used to obtain authorization information for an administrator during an introduction. Information that can be obtained includes the certificate subject name and/or the list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Example

The following example from the **show running-config**command allows you to verify that an administrative introducer using administrator authentication and authorization lists have been created:

```
Router# show running-config
Building configuration...
Current configuration : 2700 bytes
!
! Last configuration change at 01:22:26 GMT Fri Feb 4 2005
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname router
!
boot-start-marker
boot-end-marker
!
memory-size iomem 5
enable secret 5 $1$tpBS$PXnBDTIDXfX5pWa//1JX20
enable password lab
!
aaa new-model
!
!
!
aaa session-id common
!
resource manager
!
clock timezone GMT 0
ip subnet-zero
no ip routing
!
!
no ip dhcp use vrf connected
!
!
no ip cef
no ip domain lookup
ip domain name company.com
ip host router 10.3.0.6
ip host router.company.com 10.3.0.6
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
crypto pki server mycs
```

```
!
crypto pki trustpoint mycs
 revocation-check crl
 rsakeypair mycs
!
crypto pki trustpoint tti
 revocation-check crl
 rsakeypair tti
!
crypto pki trustpoint mic
 enrollment url http://router:80
 revocation-check crl
!
crypto pki trustpoint cat
 revocation-check crl
!
!
!
crypto pki certificate map cat 10
!
crypto pki certificate chain mycs
 certificate ca 01
crypto pki certificate chain tti
crypto pki certificate chain mic
 certificate 02
 certificate ca 01
crypto pki certificate chain cat
!
crypto provisioning registrar <---------- !SDP registrar device parameters!
 administrator authentication list authen-tac
 administrator authorization list author-tac
!
no crypto engine onboard 0
username qa privilege 15 password 0 lab
```

Configuring Custom Templates

Perform this task to create and configure custom templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **template http start** *URL*
5. **template http welcome** *URL*
6. **template http introduction** *URL*
7. **template http admin-introduction** *URL*
8. **template http completion** *URL*
9. **template http error** *URL*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.
Step 4	**template http start** *URL* **Example:** `Router(tti-registrar)# template http start tftp://registrar.company.com/start.html`	Directs the TTI registrar to use the custom start page template. **Note** This command is required to use the start page functionality. If this command is not issued, the welcome page is the initial communication between the introducer and the petitioner.
Step 5	**template http welcome** *URL* **Example:** `Router(tti-registrar)# template http welcome tftp://registrar.company.com/welcome.html`	(Optional) Uses a custom welcome template rather than the default template.
Step 6	**template http introduction** *URL* **Example:** `Router(tti-registrar)# template http introduction tftp://registrar.company.com/intro.html`	(Optional) Uses a custom introduction template rather than the default template.
Step 7	**template http admin-introduction** *URL* **Example:** `Router(tti-registrar)# template http admin-introduction tftp://registrar.company.com/admin-intro.html`	(Optional) Uses a custom admin-introduction template rather than the default template.
Step 8	**template http completion** *URL* **Example:**	(Optional) Uses a custom completion template rather than the default template.

	Command or Action	Purpose
	`Router(tti-registrar)# template http completion tftp://registrar.company.com/completion.html`	
Step 9	**template http error** *URL* **Example:** `Router(tti-registrar)# template http error tftp://registrar.company.com/error.html`	(Optional) Uses a custom error template rather than the default template.
Step 10	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Example

The following example shows the use of custom start, introduction, and completion templates:

template http start tftp://registrar.company.com/start.html

template http introduction tftp://registrar.company.com/intro.html

template http completion tftp://registrar.company.com/completion.html

Configuration Examples for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

Verifying the SDP Registrar Example

The following sample output from the **show running-config**command verifies that the certificate server "cs1" was configured and associated with the SDP exchange between the registrar and petitioner:

```
Router# show running-config
Building configuration...
Current configuration : 5902 bytes
!
! Last configuration change at 09:34:44 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36a
!
boot-start-marker
```

```
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$b3jz$CKquLGjFIE3AdXA2/Rl9./
enable password lab
!
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name company.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.company.com 10.23.2.131
!
!
crypto pki server cs1
 issuer-name CN=company,L=city,C=US
 hash sha1
 lifetime crl 336
 lifetime certificate 730
!
crypto pki trustpoint pki-36a
 enrollment url http://pki-36a:80
 ip-address FastEthernet0/0
 revocation-check none
!
crypto pki trustpoint cs1
 revocation-check crl
 rsakeypair cs1 2048
!
!
crypto pki certificate chain pki-36a
 certificate 03
  308201D0 30820139 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333334 345A170D 30363031 33303039 33333434 5A303A31 38301606 092A8648
  86F70D01 09081309 31302E32 332E322E 32301E06 092A8648 86F70D01 09021611
  706B692D 3336612E 63697363 6F2E636F 6D305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100AFFA 8F429618 112FAB9D 01F3352E 59DD3D2D AE67E31D
  370AC4DA 619735DF 9CF4EA13 64E4B563 C239C5F0 1578B773 07BED641 A18CA629
  191884B5 61B66ECF 4D110203 010001A3 30302E30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 141DA8B1 71652961 3F7D69F0 02903AC3 2BADB137
  C6300D06 092A8648 86F70D01 01040500 03818100 67BAE186 327CED31 D642CB39
  AD585731 95868683 B950DF14 3BCB155A 2B63CFAD B34B579C 79128AD9 296922E9
  4DEDFCAF A7B5A412 AB1FC081 09951CE3 08BFFDD9 9FB1B9DA E9AA42C8 D1049268
  C524E58F 11C6BA7F C750320C 03DFB6D4 CBB3E739 C8C76359 CE939A97 B51B3F7F
  3FF;A9D82 9CFDB6CF E2503A14 36D0A236 A1CCFEAE
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
```

```
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
crypto pki certificate chain cs1
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A02;
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
crypto provisioning registrar
 pki-server cs1
!
!
!
crypto isakmp policy 1
 hash sha
!
!
crypto ipsec transform-set test_transformset esp-aes
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.1.10
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Loopback0
 ip address 10.23.2.131 255.255.255.255
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet0/0
 ip address 10.23.2.2 255.255.255.192
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 duplex auto
```

```
 speed auto
 crypto map test_cryptomap
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 10.23.2.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.2.62
!
!
access-list 170 permit ip host 10.23.2.2 host 10.23.1.10
dialer-list 1 protocol ip permit
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

Verifying the SDP Petitioner Example

After the SDP exchange is complete, the petitioner automatically enrolls with the registrar and obtain a certificate. The following sample output through the **show running-config** command shows the automatically generated configuration, which verifies that the trustpoint is really there:

```
Router# show running-config
Building configuration...
Current configuration : 4650 bytes
!
! Last configuration change at 09:34:53 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36b
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$JYgw$060JKXgl6dERLZpU9J3gb.
enable password lab
!
```

```
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name company.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.company.com 10.23.2.131
!
!
crypto pki trustpoint tti
 enrollment url http://pki-36a.company.com:80
 revocation-check crl
 rsakeypair tti 1024
 auto-enroll 70
!
!
crypto pki certificate chain tti
 certificate 02
  308201FC 30820165 A00302012;02020102 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333333 385A170D 30363031 33303039 33333338 5A302231 20301E06 092A8648
  86F70D01 09021611 706B692D 3336622E 63697363 6F2E636F 6D30819F 300D0609
  2A864886 F70D0101 01050003 818D0030 81890281 8100E383 35584B6C 24751E2C
  F4088F06 C00BFECE 84CFF8EB 50D52044 03D14A2B 91E5A260 7D07ED24 DB599D27
  432065D9 0E459248 D7CDC15D 654E2AF6 BA27D79C 23850306 3E96C508 F311D333
  76FDDC9C A810F75C FCD10F1B 9A142F0C 338B6DB3 346D3F24 97A4B15D 0A9504E7
  1F6CB769 85E9F52B FE907AAF 63D54D66 1A715A20 D7DB0203 010001A3 30302E30
  0B060355 1D0F0404 03&#048;205A0 301F0603 551D2304 18301680 141DA8B1 71652961
  3F7D69F0 02903AC3 2BADB137 C6300D06 092A8648 86F70D01 01040500 03818100
  C5E2DA0E 4312BCF8 0396014F E18B3EE9 6C970BB7 B8FAFC61 EF849568 D546F73F
  67D2A73C 156202DC 7404A394 D6124DAF 6BACB8CF 96C3141D 109C5B0E 46F4F827
  022474ED 8B59D654 F04E31A2 C9AA1152 75A0C455 FD7EEEF5 A505A648 863EE9E6
  C361D9BD E12BBB36 16B729DF 823AD5CC 404CCE48 A4379CDC 67FF6362 0601B950
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
no crypto engine accelerator
!
!
```

```
crypto isakmp policy 1
 hash sha
!
!
crypto ipsec transform-set test_transformset esp-aes
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.2.2
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Ethernet0/0
 ip address 10.23.1.10 255.255.255.192
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 half-duplex
 crypto map test_cryptomap
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip default-gateway 10.23.1.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.1.62
!
!
access-list 170 permit ip host 10.23.1.10 host 10.23.2.2
```

```
dialer-list 1 protocol ip permit
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

Adding AAA Lists to a RADIUS or TACACS+ Server Examples

This section contains the following configuration examples:

TACACS+ AAA Server Database Example

In the following example, user information has been added to a TACACS+ AAA database. The username is "user1." The password is "cisco." Two Cisco IOS configuration template variables are configured for "user1": iosconfig1 and iosconfig2. The variables replace $1 and $2 in the configuration template file. The subject name "CN=user1, O=company, C=US" is also configured. This subject name replaces the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioner device.

```
user = user1
    password = clear "pswd"
    service=tti
        ! The certificate server inserts the following subject name to the certificate.
        set subjectname="CN=user1, O=company, C=US"
        ! Up to nine template variables may be added.
        set iosconfig1="ntp server 10.3.0.1"
        set iosconfig2="hostname user1-vpn"
```

RADIUS AAA Server Database Example

User information has been added to the RADIUS AAA server database in the following example. The username is "user1." The password is "cisco." Two Cisco IOS configuration template variables are configured for "user1": iosconfig1 and iosconfig2. The variables replace $1 and $2 in the configuration template file. The subject name "CN=user1, O=company, C=US" is also configured. This subject name replaces the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioner device.

```
user = user1
    password = clear "pswd"
    radius=company
    reply_attributes=9,1="tti:subjectname=CN=user1, O=company, C=US"
    ! Up to nine template variables may be added.
    9,1="tti:iosconfig1=ntp server 10.3.0.5"
    9,1="tti:iosconfig2=hostname user1-vpn"
```

AAA List on a TACACS+ and a RADIUS AAA Server Example

The following is a configuration example showing that AAA authentication has been configured on a TACACS+ server and that AAA authorization has been configured on a RADIUS server.

Note Authentication and authorization usually point to the same server.

```
Router(config)# tacacs-server host 10.0.0.48 key cisco
Router(config)# aaa authentication login authen-tac group tacacs+
Router(config)# radius-server host 10.0.1.49 key cisco
Router(config)# aaa authorization network author-rad group radius
```

UsingaConfigurationTemplateFile Example

You can use a different configuration template file on the basis of the introducer name. For example, if you have multiple template files for different users, each with the username in the filename, configure the following under the registrar:

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/config-$n.txt
```

In this example, the default configuration file shown in the section "Default Template for the Configuration File, on page 1258" is used because the **template config** command does not reference a CGI script.

CGI Script Example

The following example would execute a CGI script named "mysdpcgi":

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/cgi-bin/mysdpcgi post
```

The following is an example CGI script, named "mysdpcgi", that would be executed with the example **template config** command above:

```
#!/usr/bin/perl -w
# for debugging use the -debug form
# use CGI (-debug);
use CGI;
# base64 decoding is being used.
use MIME::Base64;
# The following has been commented out, but left for your information.
#
# Reading everything that has been received from stdin and writing it to the debug log to
#see what has been sent from the registrar.
#
# Remember to reset the STDIN pointer so that the normal CGI processing can get the input.
#
# print STDERR "mysdpcgi.cgi dump of stdin:\n";
# if($ENV{'REQUEST_METHOD'} eq "GET"){
#    $input_data = $ENV{'QUERY_STRING'};
# }
```

```
# else {
#    $data_length = $ENV{'CONTENT_LENGTH'};
#    $bytes_read = read(STDIN, $input_data, $data_length);
# }
# print STDERR $input_data,"\n";
# exit;

$query = new CGI;
my %av_table;
# A basic configuration file is being sent back, therefore it is being indicated as plain
# text in the command below.
print $query->header ("text/plain");
print "\n";
# For testing, parameters can be passed in so that the test applications can
# see what has been received.
#
# print STDERR "The following are the raw AV pairs mysdpcgi.cgi received:\n";
# for each $key ($query->param) {
#    print STDERR "! $key is: \n";
#    $value = $query->param($key);
#    print STDERR "! ",$value;
#    print STDERR "! \n";
#}
# The post process AV pairs are identical to those in Cisco IOS and may be used to produce
 # AV pair specific configurations as needed.
%av_table = &postprocessavpairs($query->param);
# Decoded values may be written out.
# WARNING: Some error_logs cannot handle the amount of data and will freeze.
# print STDERR "The following are the decoded AV pairs mysdpcgi.cgi received:\n";
# now write the values out
# while ( ($a, $v) = each(%av_table) ) {
#    print STDERR "$a = $v\n";
# }
# Identifying the AV pairs and specifying them in the config.
while ( ($a, $v) = each(%av_table) ) {
    if ($a eq "TTIIosRunningConfig") {
        $search = "hostname ";
        $begin = index($v, $search) + length($search);
        $end = index($v, "\n", $begin);
        $hostname = substr($v, $begin, $end - $begin);
    }
    if ($a eq "TTIIosVersion") {
        $search = "Version ";
        $begin = index($v, $search) + length($search);
        $end = index($v, "(", $begin);
        $version = substr($v, $begin, $end - $begin);
    }
}
print <<END_CONFIG;
!
! Config auto-generated by sdp.cgi
! This is for SDP testing only and is not a real config
!
!
\$t
!
\$c
!
cry pki trust Version-$version-$hostname
! NOTE: The last line of the config must be 'end' with a blank line after the end
# statement.
END_CONFIG
;
# Emulate IOS tti_postprocessavpairs functionality
```

```
sub postprocessavpairs {
    @attributes = @_;
    # Combine any AV pairs that were split apart
    $n = 0; #element index counter
    while ($attributes[$n]) {
 # see if we are at the start of a set
 if ($attributes[$n] =~ m/_0/) {
     # determine base attribute name
     $a = (split /_0/, $attributes[$n])[0];
     # set initial (partial) value
     $v = $query->param($attributes[$n]);

     # loop and pull the rest of the matching
     # attributes's values into v (would be
     # faster if we stop at first non-match)
     $c = $n+1;
     while ($attributes[$c]) {
  if ($attributes[$c] =~ m/$a/) {
      $v = $v.$query->param($attributes[$c]);
  }
  $c++;
     }

     # store in the av hash table
     $av_table{$a} = $v;
 } else {
     # store in hash table if not part of a set
     if ($attributes[$n] !~ m/_\d/) {
  $av_table{$attributes[$n]} = $query->param($attributes[$n]);
     }
 }
 $n++;
    }
    # de-base64 decode all AV pairs except userdevicename
    while ( ($a, $v) = each(%av_table) ) {
        if ($a ne "userdevicename") {
            $av_table{$a} = decode_base64($av_table{$a});
        }
    }
    return %av_table;
}
```

Note A CGI script cannot be executed without using the **post** keyword with the **template config**commandin Cisco IOS Release 12.4(6)T or a later release.

Configuring the Petitioner and Registrar for Certificate-Based Authentication Example

The following examples shows how to configure a petitioner to use the certificate issued by the trustpoint named mytrust:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning petitioner

Router(tti-petitioner)# trustpoint signing mytrust
```

```
Router(tti-petitioner)# end
```

The following example shows how to configure a registrar to verify the petitioner-signing certificate and to perform authorization lookups:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning registrar

Router(tti-registrar)# authentication trustpoint mytrust

Router(tti-registrar)# authorization login certificate

Router(tti-registrar)# authorization username subjectname all

Router(tti-registrar)# end
```

Configuring an Administrative Introducer Using Authentication and Authorization Lists Example

The following example shows how to configure an administrative introducer with the authentication list "authen-tac" and the authorization list "author-tac":

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning registrar
Router(tti-registrar)# administrator
 authentication list authen-tac
Router(tti-registrar)# administrator
 authorization list author-tac
Router(tti-registrar)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Certificate enrollment	" Configuring Certificate Enrollment for a PKI " *module*
Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " module
PKI AAA integration concepts and configuration tasks	"Configuration Revocation and Authorization of Certificates in a PKI " module

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
USB token configuration	" Storing PKI Credentials " chapter in the *Cisco IOS Security Configuration Guide: Secure Connectivity* For other 12.4T features about using SDP and USB tokens to deploy PKI credentials, see the Feature Information Table.
Integrating the iPhone, iPod touch, and iPad with enterprise systems	*Apple iPhone Enterprise Deployment Guide*
Recommended cryptographic algorithms	*Next Generation Encryption*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 152: Feature Information for SDP in a PKI

Feature Name	Releases	Feature Information
Secure Device Provisioning (SDP) Connect Template	12.4(20)T	This feature provides the ability to configure a device for Internet connectivity through a service provider.
USB Token and Secure Device Provisioning (SDP) Integration	12.4(15)T	This feature provides the ability to provision remote devices using a USB token as a mechanism to transfer credentials from one network device to a remote device through SDP. The following commands were introduced: **binary file**, **crypto key move rsa**, **template file**.
SDP Expanded Template CGI Support	12.4(6)T	This feature allows users to configure the SDP registrar to send a bootstrap configuration to the SDP petitioner based on not only the device name, but also its current Cisco IOS version and current configuration. The following command was modified by this feature: **template config.**
Secure Device Provisioning (SDP) Start Page	12.4(4)T	This feature allows users to configure their browsers to begin the TTI transaction by contacting the registrar's introduction URL through a start page. Thus, users no longer have to begin the TTI transaction from the welcome page on the petitioner. The following commands were introduced by this feature: **template http admin-introduction**, **template http completion**, **template http error**, **template http introduction**, **template http start**, **template http welcome.**
Administrative Secure Device Provisioning Introducer	12.3(14)T	This feature allows you to act as an administrative introducer to introduce a device into a PKI network and then provide a username as the device name for the record locator in the AAA database. The following commands were introduced by this feature: **administrator authentication list**, **administrator authorization list.**
Easy Secure Device Deployment	12.3(8)T	This feature introduces support for SDP, which offers a web-based enrollment interface that enables network administrators to deploy new devices in large networks. The following commands were introduced or modified: **crypto wui tti petitioner**, **crypto wui tti registrar**, **pki-server**, **template config**, **template username**, **trustpoint (tti-petitioner).**
Easy Secure Device Deployment AAA Integration	12.3(8)T	This feature integrates an external AAA database, allowing the SDP introducer to be authenticated against a AAA database instead of having to use the enable password of the local Cisco certificate server. The following commands were introduced or modified: **authentication list (tti-registrar)**, **authorization list (tti-registrar)**, **debug crypto wui template config**, **template username**.

Feature Name	Releases	Feature Information
Secure Device Provisioning (SDP) Certificate-Based Authorization	12.3(14)T	This feature allows certificates issued by other authority (CA) servers to be used for SDP introductions. The following commands were introduced by this feature: **administrator authentication list**, **administrator authorization list**
iPhone SDP	15.1(2)T	With the introduction of the Cisco IOS 15.1(2)T and Apple iPhone OS 3.0 releases, Apple iPhones are supported on Cisco IOS network devices. Cisco IOS routers use the SDP registrar to deploy iPhones so that network applications can be accessed securely through an IPSec VPN, SCEP server, and PKI certificate deployment technologies. The following commands were introduced by this feature: **match authentication trustpoint, match certificate** , **match url, mime-type**, **template location, template variable p, url-profile.**

CHAPTER 113

PKI Credentials Expiry Alerts

The PKI Credentials Expiry Alerts feature provides a warning mechanism in the form of an alert notification when a CA certificate is on the verge of expiry.

Restrictions for PKI Credentials Expiry Alerts

Alerts are not sent for the following certificates:

- Persistent or temporary self-signed certificates.
- Secure Unique Device Identifier (SUDI) certificates.
- Certificates that belong to a trustpool. Trustpools have their own expiry alerts mechanism.
- Trustpoint clones.

Information About PKI Alerts Notification

Overview of Alerts Notification

The Cisco IOS Certificate Authority (CA) server allows autoenrollment of certificates before a certificate expires to ensure the availability of certificates for applications during authentication. However, network outages, clock update problems, and overloaded CAs can impede certificate renewal, thereby resulting in subsystems going offline because no valid certificates can be used for authentication. The PKI Credentials Expiry Alerts feature provides a mechanism by which a CA client sends a notification to a syslog server when certificates are on the verge of expiry.

The notifications are sent at the following intervals:

- First notification—This is sent 60 days before the expiry of the certificate.

- Repeated notifications—After the first notification, subsequent notifications are sent every week until a week before the expiry of the certificate. In the last week, notifications are sent every day until the certificate expiry date.

The notifications are in a *warning* mode when the certificate is valid for more than a week. The notifications are in an *alert* mode when a certificate's validity is less than a week. The notifications include the following information:

- Truspoint the certificate is associated with
- Certificate type
- Serial number of the certificate
- Certificate issuer name
- Number of days remaining for the certificate to expire
- Whether the certificate is enabled with autoenrollment
- Whether a shadow certificate is available for the corresponding certificate

Note Alert notifications are sent either via the syslog server or Simple Network Management Protocol (SNMP) traps. Notifications stop when a trustpoint is configured with autoenrollment and the corresponding shadow or rollover certificate is present, and the shadow or rollover certificate's start time is either the same or earlier than the certificate's end time.

This feature cannot be disabled and requires no additional configuration tasks. The **show crypto pki timers** command is enhanced to display the timer expiry information. The following is a sample output from the **show crypto pki timers detail** command that displays the timer when a certificate is about to expire. When this timer expires, a notification is sent to the syslog server.

```
Device# show crypto pki timers detail

PKI Timers
|       14:36.150  (2019-10-30T11:33:30Z)
  |       14:36.150  (2019-10-30T11:33:30Z) SESSION CLEANUP
  |2569d23:56:19.461  (2026-11-12T11:15:13Z) SHADOW test

Expiry Alert Timers
|659d 5:56:19.599  (2021-08-19T17:15:13Z)
  |659d 5:56:19.599  (2021-08-19T17:15:13Z) ID(test)
  |2875d 4:45:18.562  (2027-09-13T16:04:12Z) CA(test)

Trustpool Timers
|3464d 9:06:48.463  (2029-04-24T20:25:42Z)
  |3464d 9:06:48.463  (2029-04-24T20:25:42Z) TRUSTPOOL
```

The following is a syslog message that is displayed on the device:

```
Device#

Dec 16 10:24:13.533: %PKI-4-CERT_EXPIRY_WARNING: ID Certificate belonging to trustpoint tp
 will expire in 60 Days 0 hours 0 mins 0 secs.
Issuer-name cn=CA
Subject-name hostname=Router
```

```
Serial-number 02
Auto-Renewal: Not Enabled
```

PKI Traps

PKI trap ease the monitoring and operations of a PKI deployment by retrieving certificate information of the devices in the network. The root device sends SNMP traps at regular intervals to the network management system (NMS) based on the threshold configured in the device. The traps are sent in the following scenarios:

- A new certificate is installed—An SNMP trap (new certificate notification) is sent to the SNMP server containing information about the certificate, such as, certificate serial number, certificate issuer name, certificate subject name, trustpoint name, certificate type, and certificate start and end date.
- A certificate is about to expire—An SNMP trap (certificate expiry notification) is sent to the SNMP server at regular intervals starting from 60 days to one week before the certificate's end date. In the week leading up to the expiration of the certificate, the trap is sent everyday. The trap contains certificate information, such as, certificate serial number, certificate issuer name, trustpoint name, certificate type, and certificate's remaining lifetime.

To enable PKI traps, use the **snmp-server enable traps pki** command.

Note If the shadow or rollover certificate's start time is later than the certificate's end time, traps are sent stating that the shadow certificate is not yet valid. However, no traps are sent if a shadow certificate available for the same trustpoint, and the shadow certificate becomes active.

Additional References for PKI Credentials Expiry Alerts

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
Security Commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 153: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 114

Configuring and Managing a Certificate Server for PKI Deployment

This module describes how to set up and manage a Cisco IOS certificate server for public key infrastructure (PKI) deployment. A certificate server embeds a simple certificate server, with limited certification authority (CA) functionality, into the Cisco software. Thus, the following benefits are provided to the user:

- Easier PKI deployment by defining default behavior. The user interface is simpler because default behaviors are predefined. That is, you can leverage the scaling advantages of PKI without all of the certificate extensions that a CA provides, thereby allowing you to easily enable a basic PKI-secured network.
- Direct integration with Cisco software.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

During copy, if running-config has both CA and ID certificates, if CA certificate is same as running-config, CA and ID are not replaced. Whereas, if CA certificate is different, then both ID and CA certificates gets cleared and new CA is re-inserted.

Prerequisites for Configuring a Certificate Server

Planning Your PKI Before Configuring the Certificate Server

Before configuring a certificate server, it is important that you have planned for and chosen appropriate values for the settings you intend to use within your PKI (such as certificate lifetimes and certificate revocation list (CRL) lifetimes). After the settings have been configured in the certificate server and certificates have been granted, settings cannot be changed without having to reconfigure the certificate server and reenrolling the peers. For information on certificate server default settings and recommended settings, see section "*Certificate Server Default Values and Recommended Values*."

Enabling an HTTP Server

The certificate server supports Simple Certificate Enrollment Protocol (SCEP) over HTTP. The HTTP server must be enabled on the router for the certificate server to use SCEP. (To enable the HTTP server, use the **ip http server** command.) The certificate server automatically enables or disables SCEP services after the HTTP server is enabled or disabled. If the HTTP server is not enabled, only manual PKCS10 enrollment is supported.

Note To take advantage of automatic CA certificate and key pair rollover functionality for all types of certificate servers, SCEP must be used as the enrollment method.

Configuring Reliable Time Services

Time services must be running on the router because the certificate server must have reliable time knowledge. If a hardware clock is unavailable, the certificate server depends on manually configured clock settings, such as Network Time Protocol (NTP). If there is not a hardware clock or the clock is invalid, the following message is displayed at bootup:

```
% Time has not been set. Cannot start the Certificate server.
```

After the clock has been set, the certificate server automatically switches to running status.

For information on manually configuring clock settings, see the module .

Restrictions for Configuring a Certificate Server

- The certificate server does not provide a mechanism for modifying the certificate request that is received from the client; that is, the certificate that is issued from the certificate server matches the requested certificate without modifications. If a specific certificate policy, such as name constraints, must be issued, the policy must be reflected in the certificate request.
-
- For validating the HTTP connection using 3rd party open SSL, the complete ISE certificate chain is sent to the device. These certificates include the ISE certificate and its issuer CA certificate. The environment data lists these certificates.

  Cisco ISE running versions 2.7.0.310 and earlier put the certificate chain in the incoming certificate list as part of environment data. In Cisco IOS XE Release 17.1.1 and earlier releases, Cisco routers do not

support multi-chain certificate downloads from ISE. Due to this, the device does not receive the ISE certificate and a TLS handshake error is displayed.

Information About Certificate Servers

RSA Key Pair and Certificate of the Certificate Server

The certificate server automatically generates a 1024-bit Rivest, Shamir, and Adelman (RSA) key pair. You must manually generate an RSA key pair if you prefer a different key pair modulus. For information on completing this task, see the section "*Generating a Certificate Server RSA Key Pair* ."

Note The recommended modulus for a certificate server RSA key pair is 2048 bits.

The certificate server uses a regular RSA key pair as its CA key. This key pair must have the same name as the certificate server. If you do not generate the key pair before the certificate server is created on the router, a general-purpose key pair is automatically generated during the configuration of the certificate server.

The CA certificate and CA key can be backed up automatically one time after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key for backup purposes.

What to Do with Automatically Generated Key Pairs

If the key pair is automatically generated, it is not marked as exportable. Thus, you must manually generate the key pair as exportable if you want to back up the CA key. For information on how to complete this task, see the section "*Generating a Certificate Server RSA Key Pair* ."

How the CA Certificate and CA Key Are Automatically Archived

At initial certificate server setup, you can enable the CA certificate and the CA key to be automatically archived so that they may be restored later if either the original copy or the original configuration is lost.

When the certificate server is turned on the first time, the CA certificate and CA key is generated. If automatic archive is also enabled, the CA certificate and the CA key is exported (archived) to the server database. The archive can be in PKCS12 or privacy-enhanced mail (PEM) format.

Note This CA key backup file is extremely important and should be moved immediately to another secured place.

- This archiving action occurs only one time. Only the CA key that is (1) manually generated and marked exportable or (2) automatically generated by the certificate server is archived (this key is marked nonexportable).
- Autoarchiving does not occur if you generate the CA key manually and mark it "nonexportable."
- In addition to the CA certificate and CA key archive file, you should also regularly back up the serial number file (.ser) and the CRL file (.crl). The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

- It is not possible to manually back up a server that uses nonexportable RSA keys or manually generated, nonexportable RSA keys. Although automatically generated RSA keys are marked as nonexportable, they are automatically archived once.

Certificate Server Database

The certificate server stores files for its own use and may publish files for other processes to use. Critical files generated by the certificate server that are needed for its ongoing operation are stored to only one location per file type for its exclusive use. The certificate server reads from and writes to these files. The critical certificate server files are the serial number file (.ser) and the CRL storage location file (.crl). Files that the certificate server writes to, but does not read from again, may be published and available for use by other processes. An example of a file that may be published is the issued certificates file (.crt).

Performance of your certificate server may be affected by the following factors, which should be considered when you choose storage options and publication options for your certificate server files.

- The storage or publish locations you choose may affect your certificate server performance. Reading from a network location takes more time than reading directly from a router's local storage device.
- The number of files you choose to store or publish to a specific location may affect your certificate server performance. The local file system may not always be suitable for a large number of files.
- The file types you choose to store or publish may affect your certificate server performance. Certain files, such as the .crl files, can become very large.

Note It is recommended that you store .ser and .crl files to your local file system and publish your .crt files to a remote file system.

Certificate Server Database File Storage

The certificate server allows the flexibility to store different critical file types to different storage locations depending on the database level set (see the **database level** command for more information). When choosing storage locations, consider the file security needed and server performance. For instance, serial number files and archive files (.p12 or .pem) might have greater security restrictions than the issued certificates file storage location (.crt) or the name file storage location (.cnm).

The table below shows the critical certificate server file types by file extension that may be stored to a specific location.

Table 154: Certificate Server Storage Critical File Types

File Extension	File Type
.ser	The main certificate server database file.
.crl	The CRL storage location.
.crt	The issued certificates storage location.
.cnm	The certificate name and expiration file storage location.

File Extension	File Type
.p12	The certificate server certificate archive file location in PKCS12 format.
.pem	The certificate server certificate archive file location in PEM format.

certificate server files may be stored to three levels of specificity:

- Default location, NVRAM
- Specified primary storage location for all critical files
- Specified storage location for specific critical file(s).

A more specific storage location setting overrides a more general storage location setting. For instance, if you have not specified any certificate server file storage locations, all certificate server files are stored to NVRAM. If you specify a storage location for the name file, only the name file is stored there; all other files continue to be stored to NVRAM. If you then specify a primary location, all files except the name file is now stored to this location, instead of NVRAM.

Note You may specify either .p12 or .pem; you cannot specify both types of archive files.

Certificate Server Database File Publication

A publish file is a copy of the original file and is available for other processes to use or for your use. If the certificate server fails to publish a file, it does cause the server to shut down. You may specify one publish location for the issued certificates file and name file and multiple publish locations for the CRL file. See the table below for files types available for publication. You may publish files regardless of the database level that is set.

Table 155: Certificate Server Publish File Types

File Extension	File Type
.crl	The CRL publish location.
.crt	The issued certificates publish location.
.cnm	The certificate name and expiration file publish location.

Trustpoint of the Certificate Server

If the certificate server also has an automatically generated trustpoint of the same name, then the trustpoint stores the certificate of the certificate server. After the router detects that a trustpoint is being used to store the certificate of the certificate server, the trustpoint is locked so that it cannot be modified.

Before configuring the certificate server you can perform the following:

- Manually create and set up this trustpoint (using the **crypto pki trustpoint**command), which allows you to specify an alternative RSA key pair (using the **rsakeypair** command).

- Specify that the initial autoenrollment key pair is generated on a specific device, such as a configured and available USB token, using the **on** command.

Note The automatically generated trustpoint and the certificate server certificate are not available for the certificate server device identity. Thus, any command-line interface (CLI) (such as the **ip http secure-trustpoint** command) that is used to specify the CA trustpoint to obtain certificates and authenticate the connecting client's certificate must point to an additional trustpoint configured on the certificate server device.

If the server is a root certificate server, it uses the RSA key pairs and several other attributes to generate a self-signed certificate. The associated CA certificate has the following key usage extensions--Digital Signature, Certificate Sign, and CRL Sign.

After the CA certificate is generated, attributes can be changed only if the certificate server is destroyed.

Note A certificate server trustpoint must not be automatically enrolled using the **auto-enroll** command. Initial enrollment of the certificate server must be initiated manually and ongoing automatic rollover functionality may be configured with the **auto-rollover** command.

Certificate Revocation Lists (CRLs)

By default, CRLs are issued once every 168 hours (1 calendar week). To specify a value other than the default value for issuing the CRL, execute the **lifetime crl** command. After the CRL is issued, it is written to the specified database location as *ca-label*.crl, where *ca-label* is the name of the certificate server.

CRLs can be distributed through SCEP, which is the default method, or a CRL distribution point (CDP), if configured and available. If you set up a CDP, use the **cdp-url** command to specify the CDP location. If the **cdp-url** command is not specified, the CDP certificate extension is not included in the certificates that are issued by the certificate server. If the CDP location is not specified, Cisco IOS PKI clients automatically request a CRL from the certificate server with a SCEP GetCRL message. The CA then returns the CRL in a SCEP CertRep message to the client. Because all SCEP messages are enveloped and signed PKCS#7 data, the SCEP retrieval of the CRL from the certificate server is costly and not highly scalable. In very large networks, an HTTP CDP provides better scalability and is recommended if you have many peer devices that check CRLs. You may specify the CDP location by a simple HTTP URL string for example,

cdp-url http://my-cdp.company.com/filename.crl

The certificate server supports only one CDP; thus, all certificates that are issued include the same CDP.

If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request and wish to use a CDP you may set up an external server to distribute CRLs and configure the CDP to point to that server. Or, you can specify a non-SCEP request for the retrieval of the CRL from the certificate server by specifying the **cdp-url** command with the URL in the following format where *cs-addr* is the location of the certificate server:

cdp-url http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL

Note If your CA is also configured as your HTTP CDP server, specify your CDP with the **cdp-url** http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL command syntax.

It is the responsibility of the network administrator to ensure that the CRL is available from the location that is specified through the **cdp-url** command.

In order to force the parser to retain the embedded question mark within the specified location, enter Ctrl-v prior to the question mark. If this action is not taken, CRL retrieval through HTTP returns an error message.

The CDP location may be changed after the certificate server is running through the **cdp-url** command. New certificates contain the updated CDP location, but existing certificates are not reissued with the newly specified CDP location. When a new CRL is issued, the certificate server uses its current cached CRL to generate a new CRL. (When the certificate server is rebooted, it reloads the current CRL from the database.) A new CRL cannot be issued unless the current CRL has expired. After the current CRL expires, a new CRL is issued only after a certificate is revoked from the CLI.

Certificate Server Error Conditions

At startup, the certificate server checks the current configuration before issuing any certificates. It reports the last known error conditions through the**show crypto pki server** command output. Example errors can include any of the following conditions:

- Storage inaccessible
- Waiting for HTTP server
- Waiting for time setting

If the certificate server experiences a critical failure at any time, such as failing to publish a CRL, the certificate server automatically enters a disabled state. This state allows the network administrator to fix the condition; thereafter, the certificate server returns to the previous normal state.

Certificate Enrollment Using a Certificate Server

A certificate enrollment request functions as follows:

- The certificate server receives the enrollment request from an end user, and the following actions occur:
  - A request entry is created in the enrollment request database with the initial state. (See the table below for a complete list of certificate enrollment request states.)
  - The certificate server refers to the CLI configuration (or the default behavior any time a parameter is not specified) to determine the authorization of the request. Thereafter, the state of the enrollment request is updated in the enrollment request database.
- At each SCEP query for a response, the certificate server examines the current request and performs one of the following actions:
  - Responds to the end user with a “pending” or “denied” state.
  - Generates and signs the appropriate certificate and stores the certificate in the enrollment request database.

If the connection of the client has closed, the certificate server waits for the client to request another certificate.

All enrollment requests transition through the certificate enrollment states that are defined in the table below. To see current enrollment requests, use the **crypto pki server request pkcs10** command.

Table 156: Certificate Enrollment Request State Descriptions

Certificate Enrollment State	Description
authorized	The certificate server has authorized the request.
denied	The certificate server has denied the request for policy reasons.
granted	The CA core has generated the appropriate certificate for the certificate request.
initial	The request has been created by the SCEP server.
malformed	The certificate server has determined that the request is invalid for cryptographic reasons.
pending	The enrollment request must be manually accepted by the network administrator.

SCEP Enrollment

All SCEP requests are treated as new certificate enrollment requests, even if the request specifies a duplicate subject name or public key pair as a previous certificate request.

Types of CA Servers Subordinate and Registration Authorities (RAs)

CA servers have the flexibility to be configured as a subordinate certificate server or an RA-mode certificate server.

Why Configure a Subordinate CA?

A subordinate certificate server provides all the same features as a root certificate server. The root RSA key pairs are extremely important in a PKI hierarchy, and it is often advantageous to keep them offline or archived. To support this requirement, PKI hierarchies allow for subordinate CAs that have been signed by the root authority. In this way, the root authority can be kept offline (except to issue occasional CRL updates), and the subordinate CA can be used during normal operation.

Why Configure an RA-Mode Certificate Server?

A certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it is forwarded to the issuing CA, and the CA automatically generates the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

An RA is the authority charged with recording or verifying some or all of the data required for the CA to issue certificates. In many cases the CA undertakes all of the RA functions itself, but where a CA operates over a wide geographical area or when there is security concern over exposing the CA to direct network access, it may be administratively advisable to delegate some of the tasks to an RA and leave the CA to concentrate on its primary tasks of signing certificates and CRLs.

CA Server Compatibility

The CA server compatibility allows the IOS CA server in RA mode to interoperate with more than one type of CA server. For more information, see "*Configuring a Certificate Server to Run in RA Mode*."

Automatic CA Certificate and Key Rollover

CAs--root CAs, subordinate CAs, and RA-mode CAs--like their clients, have certificates and key pairs with expiration dates that need to be reissued when the current certificate and key pair are about to expire. When a root CA's certificate and key pair are expiring it must generate a self-signed rollover certificate and key pair. If a subordinate CA or an RA-mode CA's certificate and key pair are expiring, it requests a rollover certificate and key pair from its superior CA, obtaining the superior CA's new self-signed rollover certificates at the same time. The CA must distribute the new CA rollover certificate and keys too all its peers. This process, called rollover, allows for continuous operation of the network while the CAs and their clients are switching from an expiring CA certificate and key pair to a new CA certificate and key pair.

Rollover relies on the PKI infrastructure requirements of trust relationships and synchronized clocks. The PKI trust relationships allow (1) the new CA certificate to be authenticated, and (2) the rollover to be accomplished automatically without the loss of security. Synchronized clocks allow the rollover to be coordinated throughout your network.

Automatic CA Certificate Rollover How It Works

The CA server must have rollover configured. All levels of CAs must be automatically enrolled and have **auto-rollover** enabled. CA clients support rollover automatically when automatically enrolled. For more information about clients and automatic rollover, see the section " Automatic Certificate Enrollment " in the chapter "Configuring Certificate Enrollment for a PKI".

After CAs have rollover enabled and their clients are automatically enrolled, there are three stages to the automatic CA certificate rollover process.

Stage One: Active CA Certificate and Key Pair Only

In stage one, there is an active CA certificate and key pair only.

Stage Two: Rollover CA Certificate and Key Pair Generation and Distribution

In stage two, the rollover CA certificate and key pair are generated and distributed. The superior CA generates a rollover certificate and key pair. After the CA successfully saves its active configuration, the CA is ready to respond to client requests for the rollover certificate and key pair. When the superior CA receives a request for the new CA certificate and key pair from a client, the CA responds by sending the new rollover CA certificate and key pair to the requesting client. The clients store the rollover CA certificate and key pair.

Note When a CA generates its rollover certificate and key pair, it must be able to save its active configuration. If the current configuration has been altered, saving of the rollover certificate and key pair does not happen automatically. In this case, the administrator must save the configuration manually or rollover information is lost.

Stage Three: Rollover CA Certificate and Key Pair Become the Active CA Certificate and Key Pair

In stage three, the rollover CA certificate and key pair become the active CA certificate and key pair. All devices that have stored a valid rollover CA certificate rename the rollover certificate to the active certificate and the once-active certificate and key pair are deleted.

After the CA certificate rollover, you may observe the following deviation from usual certificate lifetime and renewal time:

- The lifetime of the certificates issued during rollover is lower than the preconfigured value.
- In specific conditions, the renew time may be inferior to the configured percentage of the actual lifetime. The difference observed can be of up to 20% in cases where the certificate lifetime is less than one hour.

These differences are normal, and result from **jitter** (random time fluctuation) introduced by the algorithm on the Certificate server. This task is performed to avoid the hosts participating to the PKI synchronize their enrollment timer, which could result in congestion on the Certificate Server.

Note The lifetime fluctuations that occur do not affect proper functionning of the PKI, since the differences always result in a shorter lifetime, thus remaining within maximum configured lifetime for certificates.

Support for Specifying a Cryptographic Hash Function

Secure Hash Algorithm (SHA) support allows a user to specify a cryptographic hash function for Cisco IOS XE certificate servers and clients. The cryptographic hash functions that can be specified are Message Digest algorithm 5 (MD5), SHA-1, SHA-256, SHA-384, or SHA-512.

Note Cisco no longer recommends using MD5; instead, you should use SHA-256 where supported. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

See the "*Configuring a Subordinate Certificate Server*" task for more information on specifying the **hash** (ca-trustpoint) and **hash** (cs-server) commands that are used to implement this feature.

How to Set Up and Deploy a Certificate Server

Generating a Certificate Server RSA Key Pair

Perform this task to manually generate an RSA key pair for the certificate server. Manually generating a certificate server RSA key pair allows you to specify the type of key pair you want to generate, to create an exportable key pair for backup purposes, to specify the key pair storage location, or to specify the key generation location.

Note You may want to create an exportable certificate server key pair for backup, or archive purposes. If this task is not performed, the certificate server automatically generates a key pair, which is not marked as exportable.

If your device has a USB token configured and available, the USB token can be used as cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication of credentials to be performed on a USB token. The private key never leaves the USB token and is not exportable. The public key is exportable. For titles of specific documents about configuring a USB token and making it available to use as a cryptographic device, see the "Related Documents" section.

Note It is recommended that the private key be kept in a secure location and that you regularly archive the certificate server database.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
5. **crypto key import rsa** *key-label* **pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url** *url*} [**exportable**] [**on** *devicename:*] *passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**]	Generates the RSA key pair for the certificate server.

	Command or Action	Purpose
	[**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*] **Example:** `Device(config)# crypto key generate rsa label mycs exportable modulus 2048`	• The **storage** keyword specifies the key storage location. • When specifying a label name by specifying the *key-label* argument, you must use the same name for the label that you plan to use for the certificate server (through the **crypto pki server** *cs-label*command). If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used. If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the **no shutdown** command, then use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate and the private key. • By default, the modulus size of a CA RSA key is 1024 bits. The recommended modulus for a CA RSA key is 2048 bits. The range for a modulus size of a CA RSA key is from 350 to 4096 bits. • The **on** keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:). **Note** Keys created on a USB token must be 2048 bits or less.
Step 4	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **Example:** `Device(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	(Optional) Exports the generated RSA key pair. Allows you to export the generated keys.
Step 5	**crypto key import rsa** *key-label* **pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url** *url*} [**exportable**] [**on** *devicename:*] *passphrase* **Example:** `Device(config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD`	(Optional) Imports RSA key pair. To create the imported keys on a USB token, use the **on** keyword and specify the appropriate device location. If you exported the RSA keys using the **exportable** keyword and you want to change the RSA key pair to nonexportable , import the key back to the certificate server without the **exportable** keyword. The key cannot be exported again.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration.

	Command or Action	Purpose
Step 7	**show crypto key mypubkey rsa** **Example:** `Device# show crypto key mypubkey rsa`	Displays the RSA public keys of your router.

Example

The following example generates a general usage 1024-bit RSA key pair on a USB token with the label "ms2" with crypto engine debugging messages shown:

```
Device(config)# crypto key generate rsa on usbtoken0 label ms2 modulus 2048
The name for the keys will be: ms2
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:41:40.895: crypto_engine: Generate public/private keypair [OK]
Jan 7 02:44:09.623: crypto_engine: Create signature
Jan 7 02:44:10.467: crypto_engine: Verify signature
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_PUB_DECRYPT(hw)(ipsec)
```

Now, the on-token keys labeled "ms2" may be used for enrollment.

The following example shows the successful import of an encryption key to a configured and available USB tokens:

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# crypto key import rsa encryption on usbtoken0 url nvram:e password

% Importing public Encryption key or certificate PEM file...
filename [e-encr.pub]?
Reading file from nvram:e-encr.pub
% Importing private Encryption key PEM file...
Source filename [e-encr.prv]?
Reading file from nvram:e-encr.prv
% Key pair import succeeded.
```

Configuring Certificate Servers

Prerequisites for Automatic CA Certificate Rollover

When configuring a certificate server, for automatic CA certificate rollover to run successfully, the following prerequisites are applicable for your CA servers:

- Your CA server must be enabled and fully configured with a reliable time of day, an available key pair, a self-signed, valid CA certificate associated with the key pair, a CRL, an accessible storage device, and an active HTTP/SCEP server.
- CA clients must have successfully completed automatic enrollment and have autoenrollment enabled with the same certificate server.

Restrictions for Automatic CA Certificate Rollover

When configuring a certificate server, in order for automatic CA certificate rollover to run successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) is not be able to take advantage of the rollover functionality provided by SCEP.
- If you have automatic archive configured on your network and the archive fails, rollover does not occur because the certificate server does not enter the rollover state, and the rollover certificate and key pair is not automatically saved.

Configuring a Certificate Server

Perform this task to configure a certificate server and enable automatic rollover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*
5. **no shutdown**
6. **auto-rollover** [*time-period*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on your system.
Step 4	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server server-pki`	Defines a label for the certificate server and enters certificate server configuration mode. **Note** If you manually generated an RSA key pair, the *cs-label* argument must match the name of the key pair.
Step 5	**no shutdown**	(Optional) Enables the certificate server.

	Command or Action	Purpose
	Example: `Device(cs-server)# no shutdown`	**Note** Only use this command at this point if you want to use the preconfigured default functionality. That is, do not issue this command just yet if you plan to change any of the default settings as shown in the task "Configuring Certificate Server Functionality."
Step 6	**auto-rollover** [*time-period*] **Example:** `Device(cs-server)# auto-rollover 90`	(Optional) Enables the automated CA certificate rollover functionality. • *time-period*—default is 30 days.

Examples

The following example shows how to configure the certificate server "ms2" where ms2 is the label of a 2048-bit RSA key pair:

```
Device(config)# crypto pki server ms2
Device(cs-server)# no shutdown

% Once you start the server, you can no longer change some of
% the configuration.
Are you sure you want to do this? [yes/no]:
yes
% Certificate Server enabled.
Device(cs-server)# end
!
Device# show crypto pki server ms2
Certificate Server ms2:
    Status: enabled, configured
    CA cert fingerprint: 5A856122 4051347F 55E8C246 866D0AC3
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 19:44:57 GMT Oct 14 2006

CRL NextUpdate timer: 19:45:25 GMT Oct 22 2003
    Current storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

The following example shows how to enable automated CA certificate rollover on the server ms2 with the **auto-rollover** command. The **show crypto pki server**command shows that the automatic rollover has been configured on the server mycs with an overlap period of 25 days.

```
Device(config)# crypto pki server ms2
Device(cs-server)# auto-rollover 25
Device(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Exporting Certificate Server signing certificate and keys...
% Certificate Server enabled.
Device(cs-server)#
Device# show crypto pki server ms2
Certificate Server ms2:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
```

```
CA cert fingerprint:70AFECA9 211CDDCC 6AA9D7FF 3ADB03AE
Granting mode is:manual
Last certificate issued serial number:0x1
CA certificate expiration timer:00:49:26 PDT Jun 20 2008
CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
Current storage dir:nvram:
Database Level:Minimum - no cert data written to storage
Auto-Rollover configured, overlap period 25 days
Autorollover timer:00:49:26 PDT May 26 2008
```

Configuring a Subordinate Certificate Server

Perform this task to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests and to enable automatic rollover.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

Before you begin

- The root certificate server should be a Cisco IOS XE certificate server.
- For a subordinate certificate authority (CA), enrollment to the root CA or upstream CA is possible only through SCEP. The upstream CA must be online for the enrollment to the upstream CA to complete. Manual enrollment of subordinate CA to the root CA or upstream CA is not possible.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
6. **exit**
7. **crypto pki server** *cs-label*
8. **issuer name** *DN-string*
9. **mode sub-cs**
10. **auto-rollover** [*time-period*]
11. **grant auto rollover** {**ca-cert** | **ra-cert**}
12. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
13. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint sub`	Declares the trustpoint that your subordinate certificate server should use and enters ca-trustpoint configuration mode.

Step 4	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
	Example:
	`Device(ca-trustpoint)# enrollment url http://caserver.myexample.com`
	- or-
	`Device(ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80`
		Specifies the following enrollment parameters of the CA:
		• (Optional) The **mode** keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.
		• (Optional) The **retry period** keyword and *minutes* argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1.
		• (Optional) The **retry count** keyword and *number* argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10.
		• The *url* argument is the URL of the CA to which your router should send certificate requests.
		Note An IPv6 address can be added to the **http:** enrollment method. For example: http://[ipv6-address]:80. The IPv6 address must be enclosed in brackets in the URL. See the *enrollment url (ca-trustpoint)* command page for more information on the other enrollment methods that can be used.
		• (Optional) The **pem** keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.

Step 5	**hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**} **Example:** `Device(ca-trustpoint)# hash sha384`	(Optional) Specifies the hash function for the signature that the Cisco IOS XE client uses to sign its self-signed certificates. The Cisco IOS XE client uses the MD5 cryptographic hash function for self-signed certificates by default. Any of the following command algorithm keyword options can be specified to over-ride the default setting for the

	Command or Action	Purpose
		trustpoint. This setting then becomes the default cryptographic hash algorithm function for self-signed certificates by default. • **md5** —Specifies that MD5, the default hash function, is used. (No longer recommended). • **sha1** —Specifies that the SHA-1 hash function is used as the default hash algorithm for RSA keys. (No longer recommended). • **sha256** —Specifies that the SHA-256 hash function is used as the hash algorithm for Elliptic Curve (EC) 256 bit keys. • **sha384** —Specifies that the SHA-384 hash function is used as the hash algorithm for EC 384 bit keys. • **sha512** —Specifies that the SHA-512 hash function is used as the hash algorithm for EC 512 bit keys.
Step 6	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 7	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server sub`	Enables a Cisco IOS XE certificate server and enters cs-server configuration mode. **Note** The subordinate server must have the same name as the trustpoint that was created in Step 3 above.
Step 8	**issuer name** *DN-string* **Example:** `Device(cs-server)# issuer-name CN=sub CA, O=Cisco, C=us`	(Optional) Specifies the DN as the CA issuer name for the certificate server.
Step 9	**mode sub-cs** **Example:** `Device(cs-server)# mode sub-cs`	Places the PKI server into sub-certificate server mode. • Sub CA and CA relationship is supported only when all the devices on the network are of Cisco IOS XE device type. Hence a Cisco IOS XE sub CA cannot enroll to a third party CA server.
Step 10	**auto-rollover** [*time-period*] **Example:** `Device(cs-server)# auto-rollover 90`	(Optional) Enables the automated CA certificate rollover functionality. • *time-period* --default is 30 days.

	Command or Action	Purpose
Step 11	**grant auto rollover** {**ca-cert** | **ra-cert**} **Example:** `Device(cs-server)# grant auto rollover ca-cert`	(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention. • **ca-cert** --Specifies that the subordinate CA rollover certificate is automatically granted. • **ra-cert** --Specifies that the RA-mode CA rollover certificate is automatically granted. **Note** If this is the first time that a subordinate certificate server is enabled and enrolled, the certificate request must be manually granted.
Step 12	**hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**} **Example:** `Device(cs-server)# hash sha384`	(Optional) Sets the hash function for the signature that the Cisco IOS XE certificate authority (CA) uses to sign all of the certificates issued by the server. • **md5** —Specifies that MD5, the default hash function, is used. (No longer recommended). • **sha1** —Specifies that the SHA-1 hash function is used. (No longer recommended). • **sha256** —Specifies that the SHA-256 hash function is used. • **sha384** —Specifies that the SHA-384 hash function is used. • **sha512** —Specifies that the SHA-512 hash function is used.
Step 13	**no shutdown** **Example:** `Device(cs-server)# no shutdown`	Enables or reenables the certificate server. If this is the first time that a subordinate certificate server is enabled, the certificate server generates the key and obtain its signing certificate from the root certificate server.

Examples

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot your configuration as shown in the following below (Clock Not Set and Trustpoint Not Configured). Here, "ms2" refers to the label of a 2048-bit RSA key pair.

```
Router# debug crypto pki server
```

Clock Not Set

```
Router(config)# crypto pki server ms2
Router(cs-server)# mode sub-cs
```

```
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
*Jan  6 20:57:37.667: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
*Jan  6 20:57:45.303: CRYPTO_CS: starting enabling checks
*Jan  6 20:57:45.303: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
% Time has not been set. Cannot start the Certificate server
```

Trustpoint Not Configured

```
Router(config)# crypto pki server ms2
Router(cs-server)# mode sub-cs
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key or type Return to exit
Password:
Jan  6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal no shut.
Jan  6 21:03:34.309: CRYPTO_CS: enter FSM: input state initial, input signal time set.
Jan  6 21:03:34.313: CRYPTO_CS: exit FSM: new state initial.
Jan  6 21:03:34.313: CRYPTO_CS: cs config has been unlocked
Re-enter password:
Jan  6 21:03:44.413: CRYPTO_CS: starting enabling checks
Jan  6 21:03:44.413: CRYPTO_CS: associated trust point 'sub' does not exist; generated
automatically
Jan  6 21:03:44.417: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan  6 21:04:03.993: CRYPTO_CS: nvram filesystem
Jan  6 21:04:04.077: CRYPTO_CS: serial number 0x1 written.
You must specify an enrollment URL for this CA before you can authenticate it.
% Failed to authenticate the Certificate Authority
```

If the certificate server fails to obtain its signing certificate from the root certificate server, you can use the **debug crypto pki transactions**command to troubleshoot your configuration as shown in the following example:

```
Router# debug crypto pki transactions
Jan  6 21:07:00.311: CRYPTO_CS: enter FSM: input state initial, input signal time set
Jan  6 21:07:00.311: CRYPTO_CS: exit FSM: new state initial
Jan  6 21:07:00.311: CRYPTO_CS: cs config has been unlocked no sh
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan  6 21:07:03.535: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan  6 21:07:10.619: CRYPTO_CS: starting enabling checks
Jan  6 21:07:10.619: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan  6 21:07:20.535: %SSH-5-ENABLED: SSH 1.99 has been enabled
Jan  6 21:07:25.883: CRYPTO_CS: nvram filesystem
Jan  6 21:07:25.991: CRYPTO_CS: serial number 0x1 written.
Jan  6 21:07:27.863: CRYPTO_CS: created a new serial file.
Jan  6 21:07:27.863: CRYPTO_CS: authenticating the CA 'sub'
Jan  6 21:07:27.867: CRYPTO_PKI: Sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message=sub HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Cisco PKI)
Jan 6 21:07:27.867: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:27.871: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6 Certificate has the
following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2
% Do you accept this certificate? [yes/no]:
Jan  6 21:07:30.879: CRYPTO_PKI: http connection opened
```

```
Jan 6 21:07:30.903: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:30 GMT
Server: server-IOS
Content-Type: application/x-x509-ca-cert
Expires: Thu, 06 Jan 2005 21:07:30 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:30 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none
Content-Type indicates we have received a CA certificate.
Jan  6 21:07:30.903: Received 507 bytes from server as CA certificate:
Jan  6 21:07:30.907: CRYPTO_PKI: transaction GetCACert completed
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.927: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()
Jan  6 21:07:30.927: CRYPTO_PKI: trustpoint sub authentication status = 0 y Trustpoint CA
certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
Router (cs-server)#
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:52.460: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan 6 21:07:54.348: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan 6 21:07:54.352: CRYPTO_CS: exit FSM: new state check failed
Jan 6 21:07:54.352: CRYPTO_CS: cs config has been locked
Jan 6 21:07:54.356: CRYPTO_PKI: transaction PKCSReq completed
Jan 6 21:07:54.356: CRYPTO_PKI: status:
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 1BA027DB 1C7860C7
EC188F65 64356C80
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 840DB52C E17614CB
0C7BE187 0DFC884D D32CAA75
Jan 6 21:07:56.508: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:56.508: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:07:56.516: CRYPTO_PKI: http connection opened
Jan 6 21:07:59.136: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:07:57 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:07:57 GMT
  Last-Modified: Thu, 06 Jan 2005 21:07:57 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
Jan 6 21:07:59.324: The PKCS #7 message has 1 verified signers.
Jan 6 21:07:59.324: signing cert: issuer=cn=root1
Jan 6 21:07:59.324: Signed Attributes:
Jan 6 21:07:59.328: CRYPTO_PKI: status = 102: certificate request pending
Jan 6 21:08:00.788: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:08:00.788: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:08:00.796: CRYPTO_PKI: http connection opened
Jan 6 21:08:11.804: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:08:11.804: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:08:01 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:08:01 GMT
  Last-Modified: Thu, 06 Jan 2005 21:08:01 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
```

```
Jan  6 21:08:11.992: The PKCS #7 message has 1 verified signers.
Jan  6 21:08:11.992: signing cert: issuer=cn=root1
Jan  6 21:08:11.996: Signed Attributes:
Jan  6 21:08:11.996: CRYPTO_PKI: status = 102: certificate request pending
Jan  6 21:08:21.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:31.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:41.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:51.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:01.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial, 1
Jan  6 21:09:11.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial for session: 0
Jan  6 21:09:11.996: CRYPTO_PKI: can not resolve server name/IP address
Jan  6 21:09:11.996: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan  6 21:09:12.024: CRYPTO_PKI: http connection opened% Exporting Certificate Server signing
 certificate and keys...
Jan  6 21:09:14.784: CRYPTO_PKI: received msg of 1611 bytes
Jan  6 21:09:14.784: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:09:13 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:09:13 GMT
  Last-Modified: Thu, 06 Jan 2005 21:09:13 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
Jan  6 21:09:14.972: The PKCS #7 message has 1 verified signers.
Jan  6 21:09:14.972: signing cert: issuer=cn=root1
Jan  6 21:09:14.972: Signed Attributes:
Jan  6 21:09:14.976: CRYPTO_PKI: status = 100: certificate is granted
Jan  6 21:09:15.668: The PKCS #7 message contains 1 certs and 0 crls.
Jan  6 21:09:15.688: Newly-issued Router Cert: issuer=cn=root serial=2
Jan  6 21:09:15.688: start date: 21:08:03 GMT Jan 6 2005
Jan  6 21:09:15.688: end date: 21:08:03 GMT Jan 6 2006
Jan  6 21:09:15.688: Router date: 21:09:15 GMT Jan 6 2005
Jan  6 21:09:15.692: Received router cert from CA
Jan  6 21:09:15.740: CRYPTO_CA: certificate not found
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.748: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 21:09:15.748: CRYPTO_CS: starting enabling checks
Jan  6 21:09:15.748: CRYPTO_CS: nvram filesystem
Jan  6 21:09:15.796: CRYPTO_CS: found existing serial file.
Jan  6 21:09:15.820: CRYPTO_CS: old router cert flag 0x4
Jan  6 21:09:15.820: CRYPTO_CS: new router cert flag 0x44
Jan  6 21:09:18.432: CRYPTO_CS: DB version 1
Jan  6 21:09:18.432: CRYPTO_CS: last issued serial number is 0x1
Jan  6 21:09:18.480: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 21:09:18.480: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 21:09:18.532: CRYPTO_CS: SCEP server started
Jan  6 21:09:18.532: CRYPTO_CS: exit FSM: new state enabled
Jan  6 21:09:18.536: CRYPTO_CS: cs config has been locked
Jan  6 21:09:18.536: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
```

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot the progress of an enrollment. This command can also be used to debug the root CA (turn it on at the root CA).

Configuring a Certificate Server to Run in RA Mode

The certificate server can act as an RA for a CA or another third party CA. Read the details in Step 8 for more information about the **transparent** keyword option if a third-party CA is used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **subject-name** *x.500-name*
6. **exit**
7. **crypto pki server** *cs-label*
8. **mode ra** [**transparent**]
9. **auto-rollover** [*time-period*]
10. **grant auto rollover** {**ca-cert** | **ra-cert**}
11. **no shutdown**
12. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint ra-server`	Declares the trustpoint that your RA mode certificate server should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Device(ca-trustpoint)# enrollment url http://ca-server.company.com`	Specifies the enrollment URL of the issuing CA certificate server (root certificate server).
Step 5	**subject-name** *x.500-name* **Example:** `Device(ca-trustpoint)# subject-name cn=ioscs RA`	Specifies the subject name the RA uses. **Note** Include "cn=ioscs RA" or "ou=ioscs RA" in the subject name so that the issuing CA certificate server can recognize the RA (see Step 7 below).

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 7	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server ra-server`	Enables a certificate server and enters cs-server configuration mode. **Note** The certificate server must have the same name as the trustpoint that was created in Step 3 above.
Step 8	**mode ra** [**transparent**] **Example:** `Device(cs-server)# mode ra`	Places the PKI server into RA certificate server mode. Use the **transparent** keyword to allow the CA server in RA mode to interoperate with more than one type of CA server. When the **transparent** keyword is used, the original PKCS#10 enrollment message is not re-signed and is forwarded unchanged. This enrollment message makes the IOS RA certificate server work with CA servers like the Microsoft CA server.
Step 9	**auto-rollover** [*time-period*] **Example:** `Device(cs-server)# auto-rollover 90`	(Optional) Enables the automatic CA certificate rollover functionality. • *time-period* --default is 30 days.
Step 10	**grant auto rollover** {**ca-cert** | **ra-cert**} **Example:** `Device(cs-server)# grant auto rollover ra-cert`	(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention. • **ca-cert** --Specifies that the subordinate CA rollover certificate is automatically granted. • **ra-cert** --Specifies that the RA-mode CA rollover certificate is automatically granted. If this is the first time that a subordinate certificate server is enabled and enrolled, the certificate request must be manually granted.
Step 11	**no shutdown** **Example:** `Device(cs-server)# no shutdown`	Enables the certificate server. **Note** After this command is issued, the RA automatically enrolls with the root certificate server. After the RA certificate has been successfully received, you must issue the **no shutdown** command again, which reenables the certificate server.
Step 12	**no shutdown** **Example:**	Reenables the certificate server.

	Command or Action	Purpose
	`Device(cs-server)# no shutdown`	

Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server

Perform the following steps on the router that is running the issuing certificate server; that is, configure the root certificate server that is delegating enrollment tasks to the RA mode certificate server.

Note Granting enrollment requests for an RA is essentially the same process as granting enrollment requests for client devices--except that enrollment requests for an RA are displayed in the section "RA certificate requests" of the command output for the **crypto pki server info-requests** command.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **info requests**
3. **crypto pki server** *cs-label* **grant** *req-id*
4. **configure terminal**
5. **crypto pki server** *cs-label*
6. **grant ra-auto**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **info requests** **Example:** `Device# crypto pki server root-server info requests`	Displays the outstanding RA certificate request. **Note** This command is issued on the router that is running the issuing certificate server.
Step 3	**crypto pki server** *cs-label* **grant** *req-id* **Example:** `Device# crypto pki server root-server grant 9`	Grants the pending RA certificate request. **Note** Because the issuing certificate server delegates the enrollment request verification task to the RA, you must pay extra attention to the RA certificate request before granting it.
Step 4	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 5	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server root-server`	Enables a certificate server and enters cs-server configuration mode.
Step 6	**grant ra-auto** **Example:** `Device(cs-server)# grant ra-auto`	(Optional) Specifies that all enrollment requests from an RA are to be granted automatically. **Note** For the **grant ra-auto** command to work, you have to include "cn=ioscs RA" or "ou=ioscs RA" in the subject name of the RA certificate. (See Step 2 above.)

What to Do Next

After you have configured a certificate server, you can use the preconfigured default values or specify values through the CLI for the functionality of the certificate server. If you choose to specify values other than the defaults, see the following section, "*Configuring Certificate Server Functionality* ."

Configuring Certificate Server Functionality

After you have enabled a certificate server and are in certificate server configuration mode, use any of the steps in this task to configure basic certificate server functionality values other than the default values.

Certificate Server Default Values and Recommended Values

The default values for a certificate server are intended to address a relatively small network (of about ten devices). For example, the database settings are minimal (through the **database level minimal**command) and the certificate server handles all CRL requests through SCEP. For larger networks, it is recommended that you use either the database setting "names" or "complete" (as described in the **database level** command) for possible audit and revocation purposes. Depending on the CRL checking policy, you should also use an external CDP in a larger network.

Certificate Server File Storage and Publication Locations

You have the flexibility to store file types to different storage and publication locations.

SUMMARY STEPS

1. **database url** *root-url*
2. **database url** {**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url*
3. **database url** {**cnm** | **crl** | **crt**} **publish** *root-url*
4. **database level** {**minimal** | **names** | **complete**}
5. **database username** *username* [**password** [*encr-type*] *password*]
6. **database archive** {**pkcs12** | **pem**}[**password** *encr-type*] *password*]
7. **issuer-name** *DN-string*
8. **lifetime** {**ca-certificate** | **certificate**} *time*
9. **lifetime crl** *time*

10. **lifetime enrollment-request** *time*
11. **cdp-url** *url*
12. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**database url** *root-url* **Example:** `Device(cs-server)# database url tftp://cert-svr-db.company.com`	Specifies the primary location where database entries for the certificate server are written. If this command is not specified, all database entries are written to NVRAM.
Step 2	**database url** {**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url* **Example:** `Device(cs-server)# database url ser nvram:`	Specifies certificate server critical file storage location by file type. **Note** If this command is not specified, all critical files are stored to the primary location if specified. If the primary location is not specified, all critical files are stored to NVRAM.
Step 3	**database url** {**cnm** | **crl** | **crt**} **publish** *root-url* **Example:** `Device(cs-server)# database url crl publish tftp://csdb_specific_crl_files.company.com`	Specifies certificate server publish location by file type. **Note** If this command is not specified, all publish files are stored to the primary location if specified. If the primary location is not specified, all publish files are stored to NVRAM.
Step 4	**database level** {**minimal** | **names** | **complete**} **Example:** `Device(cs-server)# database level complete`	Controls what type of data is stored in the certificate enrollment database. • **minimal** --Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names** --In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete** --In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is issued, you should also specify an external TFTP server in which to store the data through the **database url** command.

	Command or Action	Purpose
Step 5	**database username** *username* [**password** [*encr-type*] *password*] **Example:** `Device(cs-server)# database username user password PASSWORD`	(Optional) Sets a username and password when a user is required to access a primary certificate enrollment database storage location.
Step 6	**database archive** {**pkcs12** | **pem**}[**password** *encr-type*] *password*] **Example:** `Device(cs-server)# database archive pem`	(Optional) Sets the CA key and CA certificate archive format and password to encrypt the file. The default value is **pkcs12**, so if this subcommand is not configured, autoarchiving continues, and the PKCS12 format is used. • The password is optional. If it is not configured, you are prompted for the password when the server is turned on for the first time. **Note** It is recommended that you remove the password from the configuration after the archive is finished.
Step 7	**issuer-name** *DN-string* **Example:** `Device(cs-server)# issuer-name my-server`	(Optional) Sets the CA issuer name to the specified distinguished name (*DN-string*). The default value is as follows: **issuer-name cn**={*cs-label* }.
Step 8	**lifetime** {**ca-certificate** | **certificate**} *time* **Example:** `Device(cs-server)# lifetime certificate 888`	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.
Step 9	**lifetime crl** *time* **Example:** `Device(cs-server)# lifetime crl 333`	(Optional) Defines the lifetime, in hours, of the CRL that is used by the certificate server. Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).
Step 10	**lifetime enrollment-request** *time* **Example:** `Device(cs-server)# lifetime enrollment-request 888`	(Optional) Specifies how long an enrollment request should stay in the enrollment database before being removed. Maximum lifetime is 1000 hours.
Step 11	**cdp-url** *url* **Example:** `Device(cs-server)# cdp-url http://my-cdp.company.com`	(Optional) Defines the CDP location to be used in the certificates that are issued by the certificate server. • The URL must be an HTTP URL.

	Command or Action	Purpose
		If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request, use the following URL format: http://server.company.com/certEnroll/filename.crl Or, if your Cisco IOS certificate server is also configured as your CDP, use the following URL format http://cs-addr/cgi-bin/pkiclient.exe?operation=GetCRL where *cs-addr* is the location of the certificate server. In order to force the parser to retain the embedded question mark within the specified location, enter Ctrl-v prior to the question mark. If this action is not taken, CRL retrieval through HTTP returns an error message. **Note** Although this command is optional, it is strongly recommended for any deployment scenario.
Step 12	**no shutdown** **Example:** `Device(cs-server)# no shutdown`	Enables the certificate server. You should issue this command only after you have completely configured your certificate server.

Examples

The following example shows how to configure a CDP location where the PKI clients do not support SCEP GetCRL requests:

```
Device(config)# crypto pki server aaa
Device(cs-server)# database level minimum
Device(cs-server)# database url tftp://10.1.1.1/username1/
Device(cs-server)# issuer-name CN=aaa
Device(cs-server)# cdp-url http://server.company.com/certEnroll/aaa.crl
```

After a certificate server has been enabled on a router, the **show crypto pki server**command displays the following output:

```
Device# show crypto pki server

    Certificate Server status:enabled, configured
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:19:31:15 PST Nov 17 2006
    CRL NextUpdate timer:19:31:15 PST Nov 25 2003
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
```

Working with Automatic CA Certificate Rollover

Starting Automated CA Certificate Rollover Immediately

Use this task to initiate the automated CA certificate rollover process immediately on your root CA server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki server** *cs-label* **rollover** [**cancel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki server** *cs-label* **rollover** [**cancel**] **Example:** `Device(config)# crypto pki server mycs rollover`	Immediately starts the CA certificate rollover process by generating a shadow CA certificate. To delete the CA certificate rollover certificate and keys, use the **cancel** keyword.

Requesting a Certificate Server Client Rollover Certificate

Use this task to request a certificate server client's rollover certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki server** *cs-label* **rollover request pkcs10 terminal**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**crypto pki server** *cs-label* **rollover request pkcs10 terminal** **Example:** `Device(config)# crypto pki server mycs rollover request pkcs10 terminal`	Requests a client rollover certificate from the server.

Example

The following example shows a rollover certificate request being inputted into the server:

```
Device# crypto pki server mycs rollover request pkcs10 terminal

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE REQUEST-----
MIIBUTCBuwIBADASMRAwDgYDVQQDEwdOZXdSb290MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDMHeev1ERSs320zbLQQk+3lhV/R2HpYQ/iM6uT1jkJf5iy0UPR
wF/X16yUNmG+ObiGiW9fsASF0nxZw+fO7d2X2yh1PakfvF2wbP27C/sgJNOw9uPf
sBxEc40Xe0d5FMh0YKOSAShfZYKOflnyQR2Drmm2x/33QGol5QyRvjkeWQIDAQAB
oAAwDQYJKoZIhvcNAQEEBQADgYEALM90r4d79X6vxhD0qjuYJXfBCOvv4FNyFsjr
aBS/y6CnNVYySF8UBUohXYIGTWf4I4+sj6i8gYfoFUW1/L82djS18TLrUr6wpCOs
RqfAfps7HW1e4cizOfjAUU+C7lNcobCAhwF1o6q2nIEjpQ/2yfK9O7sb3SCJZBfe
eW3tyCo=
-----END CERTIFICATE REQUEST-----
```

Exporting a CA Rollover Certificate

Use this task to export a CA rollover certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki export** *trustpoint* **pem** {**terminal** | **url** *url*} [**rollover**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki export** *trustpoint* **pem** {**terminal** | **url** *url*} [**rollover**]	Exports a CA shadow certificate.

	Command or Action	Purpose
	Example: `Device(config)# crypto pki export mycs pem terminal rollover`	

Maintaining Verifying and Troubleshooting the Certificate Server Certificates and the CA

Managing the Enrollment Request Database

SCEP supports two client authentication mechanisms--manual and preshared key. Manual enrollment requires the administrator at the CA server to specifically authorize the enrollment requests; enrollment using preshared keys allows the administrator to preauthorize enrollment requests by generating a one-time password (OTP).

Use any of the optional steps within this task to help manage the enrollment request database by performing functions such as specifying enrollment processing parameters that are to be used by SCEP and by controlling the run-time behavior or the certificate server.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **grant** {**all** | *req-id*}
3. **crypto pki server** *cs-label* **reject** {**all** | *req-id*}
4. **crypto pki server** *cs-label* **password generate** *minutes*
5. **crypto pki server** *cs-label* **revoke** *certificate-serial-number*
6. **crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**base64**| **pem**
7. **show crypto pki server** *cs-label* **crl**
8. **show crypto pki server** *cs-label* **requests**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **grant** {**all**	*req-id*} **Example:** `Device# crypto pki server mycs grant all`	Grants all or specific SCEP requests.
Step 3	**crypto pki server** *cs-label* **reject** {**all**	*req-id*} **Example:** `Device# crypto pki server mycs reject all`	Rejects all or specific SCEP requests.
Step 4	**crypto pki server** *cs-label* **password generate** *minutes* **Example:**	Generates a OTP for SCEP requests.

	Command or Action	Purpose
	`Device# crypto pki server mycs password generate 75`	• *minutes* --Length of time, in minutes, that the password is valid. Valid values range from 1 to 1440 minutes. The default is 60 minutes. **Note** Only one OTP is valid at a time; if a second OTP is generated, the previous OTP is no longer valid.
Step 5	**crypto pki server** *cs-label* **revoke** *certificate-serial-number* **Example:** `Device# crypto pki server mycs revoke 3`	Revokes a certificate on the basis of its serial number. • *certificate-serial-number* --One of the following options: • A string with a leading 0x, which is treated as a hexadecimal value • A string with a leading 0 and no x, which is treated as octal • All other strings, which are treated as decimal
Step 6	**crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**base64**| **pem** **Example:** `Device# crypto pki server mycs request pkcs10 terminal pem`	Manually adds either a base64-encoded or PEM-formatted PKCS10 certificate enrollment request to the request database. After the certificate is granted, it is displayed on the console terminal using base64 encoding. • **pem** --Specifies the certificate that is returned with PEM headers automatically added to the certificate after the certificate is granted, regardless of whether PEM headers were used in the request. • **base64** --Specifies the certificate that is returned without privacy-enhanced mail (PEM) headers, regardless of whether PEM headers were used in the request.
Step 7	**show crypto pki server** *cs-label* **crl** **Example:** `Device# show crypto pki server mycs crl`	Displays information regarding the status of the current CRL.
Step 8	**show crypto pki server** *cs-label* **requests** **Example:** `Device# show crypto pki server mycs requests`	Displays all outstanding certificate enrollment requests.

Removing Requests from the Enrollment Request Database

After the certificate server receives an enrollment request, the server can leave the request in a pending state, reject it, or grant it. The request stays in the enrollment request database for 1 week until the client polls the certificate server for the result of the request. If the client exits and never polls the certificate server, you can remove either individual requests or all requests from the database.

Use this task to remove requests from the database and allow the server to be returned to a clean slate with respect to the keys and transaction IDs. Also, you can use this task to help troubleshoot a SCEP client that may not be behaving properly.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **remove** {**all** | *req-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **remove** {**all**	*req-id*} **Example:** `Device# crypto pki server mycs remove 15`	Removes enrollment requests from the enrollment request database.

Deleting a Certificate Server

Users can delete a certificate server from the PKI configuration if they no longer want it on the configuration. Typically, a subordinate certificate server or an RA is being deleted. However, users may delete a root certificate server if they are moving it to another device through the archived RSA keys.

Perform this task to delete a certificate server from your PKI configuration.

Note When a certificate server is deleted, the associated trustpoint and key are also deleted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto pki server** *cs-label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**no crypto pki server** *cs-label* **Example:** `Device(config)# no crypto pki server mycs`	Deletes a certificate server and associated trustpoint and key.

Verifying and Troubleshooting Certificate Server and CA Status

Use any of the following optional steps to verify the status of the certificate server or the CA.

SUMMARY STEPS

1. **enable**
2. **debug crypto pki server**
3. **dir** *filesystem* :

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto pki server** **Example:** `Device# debug crypto pki server`	Enables debugging for a crypto PKI certificate server. • This command can be used for monitoring the progress of an enrollment and for troubleshooting if the certificate server fails to respond or if the certificate server has trouble handling the request that has been configured.
Step 3	**dir** *filesystem* : **Example:** `Device# dir slot0:`	Displays a list of files on a file system. • This command can be used to verify the certificate server autoarchived file if the **database url** command was entered to point to a local file system. You should be able to at least see "*cs-label* .ser" and "*cs-label* .crl" files in the database.

Verifying CA Certificate Information

To obtain information relating to the CA certificates including the certificate server rollover process, rollover certificates, and timers, you may use any of the following commands.

Note These commands are not exclusive to shadow certificate information. If no shadow certificate exists, the following commands display the active certificate information.

SUMMARY STEPS

1. **crypto pki certificate chain**
2. **crypto pki server info requests**
3. **show crypto pki certificates**
4. **show crypto pki server**
5. **show crypto pki trustpoints**

DETAILED STEPS

Step 1 crypto pki certificate chain

Example:

```
Device(config)# crypto pki certificate chain mica

certificate 06
certificate ca 01
! This is the peer's shadow PKI certificate.
certificate rollover 0B
! This is the CA shadow PKI certificate
certificate rollover ca 0A
```

Displays the certificate chain details and to distinguish the current active certificate from the rollover certificate in the certificate chain. The following example shows a certificate chain with an active CA certificate and a shadow, or rollover, certificate:

Step 2 crypto pki server info requests

Example:

```
Device# crypto pki server myca info requests

Enrollment Request Database:
RA certificate requests:
  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
RA rollover certificate requests:
  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------
Router certificates requests:
  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
1      pending    A426AF07FE3A4BB69062E0E47198E5BF hostname=client
  Router rollover certificates requests:
  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------
  2      pending    B69062E0E47198E5BFA426AF07FE3A4B hostname=client
```

Displays all outstanding certificate enrollment requests. The following example shows the output for shadow PKI certificate information requests:

Step 3 show crypto pki certificates

Example:

```
Device# show crypto pki certificates

Certificate
  Subject Name
    Name: myrouter.example.com
```

```
    IP Address: 192.0.2.1
    Serial Number: 04806682
  Status: Pending
  Key Usage: General Purpose
    Fingerprint: 428125BD A3419600 3F6C7831 6CD8FA95 00000000
CA Certificate
  Status: Available
  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F
  Key Usage: Not Set
```

Displays information about the certificate, the certification authority certificate, shadow certificates, and any registration authority certificates. The following example displays the certificate of the router and the certificate of the CA. There is no shadow certificate available. A single, general-purpose RSA key pair was previously generated, and a certificate was requested but not received for that key pair. Note that the certificate status of the router shows "Pending." After the router receives its certificate from the CA, the Status field changes to "Available" in the **show** output.

Step 4 **show crypto pki server**

Example:

```
Device# show crypto pki server

Certificate Server routercs:
   Status: enabled, configured
   Issuer name: CN=walnutcs
   CA cert fingerprint: 800F5944 74337E5B C2DF6C52 9A7B1BDB
   Granting mode is: auto
   Last certificate issued serial number: 0x7
   CA certificate expiration timer: 22:10:29 GMT Jan 29 2007
   CRL NextUpdate timer: 21:50:56 GMT Mar 5 2004
   Current storage dir: nvram:
   Database Level: Minimum - no cert data written to storage
Rollover status: available for rollover
    Rollover CA cert fingerprint: 6AAF5944 74227A5B 23DF3E52 9A7F1FEF
    Rollover CA certificate expiration timer: 22:10:29 GMT Jan 29 2017
```

Displays the current state and configuration of the certificate server. The following example shows that the certificate server "routercs" has rollover configured. The CA auto-rollover time has occurred and the rollover, or shadow, PKI certificate is available. The status shows the rollover certificate fingerprint and rollover CA certificate expiration timer information.

Step 5 **show crypto pki trustpoints**

Example:

```
Device# show crypto pki trustpoints

Trustpoint vpn:
   Subject Name:
   cn=Cisco SSL CA
   o=Cisco Systems
   Serial Number: 0FFEBBDC1B6F6D9D0EA7875875E4C695
   Certificate configured.
   Rollover certificate configured.
   Enrollment Protocol:
   SCEPv1, PKI Rollover
```

Displays the trustpoints that are configured in the device. The following output shows that a shadow CA certificate is available and shows the SCEP capabilities reported during the last enrollment operation:

Configuration Examples for Using a Certificate Server

Example: Configuring Specific Storage and Publication Locations

The following example shows the configuration of a minimal local file system, so that the certificate server can respond quickly to certificate requests. The .ser and .crl files are stored on the local system for fast access, and a copy of all of the .crt files are published to a remote location for long-term logging.

```
crypto pki server myserver
     !Pick your database level.
     database level minimum
     !Specify a location for the .crt files that is different than the default local
!Cisco IOS file system.
     database url crt publish http://url username user1 password secret
```

Note Free space on the local file system should be monitored, in case the .crl file becomes too large.

The following example shows the configuration of a primary storage location for critical files, a specific storage location for the critical file serial number file, the main certificate server database file, and a password protected file publication location for the CRL file:

```
Device(config)# crypto pki server mycs
Device(cs-server)# database url ftp://cs-db.company.com

!
% Server database url was changed. You need to move the
% existing database to the new location.
!
Device(cs-server)# database url ser nvram:
Device(cs-server)# database url crl publish ftp://crl.company.com username myname password
 mypassword
Device(cs-server)# end
```

The following output displays the specified primary storage location and critical file storage locations specified:

```
Device# show

Sep  3 20:19:34.216: %SYS-5-CONFIG_I: Configured from console by user on console
Device# show crypto pki server

Certificate Server mycs:
     Status: disabled
     Server's configuration is unlocked  (enter "no shut" to lock it)
     Issuer name: CN=mycs
     CA cert fingerprint: -Not found-
     Granting mode is: manual
     Last certificate issued serial number: 0x0
     CA certificate expiration timer: 00:00:00 GMT Jan 1 1970
     CRL not present.
     Current primary storage dir: ftp://cs-db.company.com
     Current storage dir for .ser files: nvram:
    Database Level: Minimum - no cert data written to storage The following output displays
 all storage and publication locations. The serial number file (.ser) is stored in NVRAM.
```

```
The CRL file will be published to ftp://crl.company.com with a username and password. All
other critical files will be stored to the primary location, ftp://cs-db.company.com.

Device# show running-config

   section crypto pki server
   crypto pki server mycs shutdown database url ftp://cs-db.company.com
   database url crl publish ftp://crl.company.com username myname password 7
12141C0713181F13253920
   database url ser nvram:
Device#
```

Example: Removing Enrollment Requests from the Enrollment Request Database

The following examples show both the enrollment requests that are currently in the enrollment request database and the result after one of the enrollment requests has been removed from the database.

Example: Enrollment Request Currently in the Enrollment Request Database

The following example shows that the **crypto pki server info requests** command has been used to display the enrollment requests that are currently in the Enrollment Request Database:

```
Device# crypto pki server myserver info requests

Enrollment Request Database:
RA certificate requests:
ReqID    State      Fingerprint                                SubjectName
--------------------------------------------------------------------------
Router certificates requests:
ReqID    State      Fingerprint                                SubjectName
--------------------------------------------------------------------------
2        pending    1B07F3021DAAB0F19F35DA25D01D8567      hostname=host1.company.com
1        denied     5322459D2DC70B3F8EF3D03A795CF636      hostname=host2.company.com
```

Example: crypto pki server remove Command Used to Remove One Enrollment Request

The following example shows that the **crypto pki server remove** command has been used to remove Enrollment Request 1:

```
Device# crypto pki server myserver remove 1
```

Example: Enrollment Request Database After the Removal of One Enrollment Request

The following example shows the result of the removal of Enrollment Request 1 from the Enrollment Request Database:

```
Device# crypto pki server mycs info requests

Enrollment Request Database:
RA certificate requests:
ReqID    State     Fingerprint                        SubjectName
--------------------------------------------------------------------
Router certificates requests:
ReqID    State     Fingerprint                        SubjectName
--------------------------------------------------------------------
2        pending   1B07F3021DAAB0F19F35DA25D01D8567   hostname=host1.company.com
```

Example: Autoarchiving the Certificate Server Root Keys

The following output configurations and examples show what you might see if the **database archive** command has not been configured (that is, configured using the default value); if the **database archive** command has been configured to set the CA certificate and CA key archive format as PEM, without configuring a password; and if the **database archive** command has been configured to set the CA certificate and CA key archive format as PKCS12, with a password configured. The last example is sample content of a PEM-formatted archive file. The following example, "ms2" refers to the label of a 2048-bit key pair.

Example: database archive Command Not Configured

Note The default is PKCS12, and the prompt for the password appears after the **no shutdown** command has been issued.

```
Device(config)# crypto pki server ms2
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Device(cs-server)# end
Device# dir nvram:

Directory of nvram:/
  125  -rw-        1693              <no date>  startup-config
  126  ----           5              <no date>  private-config
    1  -rw-          32              <no date>  myserver.ser
    2  -rw-         214              <no date>  myserver.crl
! Note the next line, which indicates PKCS12 format.
    3  -rw-        1499              <no date>  myserver.p12
```

Example" database archive Command and pem Keyword Configured

Note The prompt for the password appears after the **no shutdown** command has been issued.

```
Device(config)# crypto pki server ms2
Device(cs-server)# database archive pem
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
!Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
```

```
Device(cs-server)# end
Device# dir nvram

Directory of nvram:/
  125  -rw-        1693            <no date>  startup-config
  126  ----           5            <no date>  private-config
    1  -rw-          32            <no date>  myserver.ser
    2  -rw-         214            <no date>  myserver.crl
! Note the next line showing that the format is PEM.
    3  -rw-        1705            <no date>  myserver.pem
```

Example: database archive Command and pkcs12 Keyword (and Password) Configured

Note When the password is entered, it is encrypted. However, it is recommended that you remove the password from the configuration after the archive has finished.

```
Device(config)# crypto pki server ms2
Device(cs-server)# database archive pkcs12 password cisco123
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note that you are not being prompted for a password.
% Certificate Server enabled.
Device(cs-server)# end
Device# dir nvram:

Directory of nvram:/
   125   -rw-         1693             <no date>   startup-config
   126   ----            5             <no date>   private-config
     1   -rw-           32             <no date>   myserver.ser
     2   -rw-          214             <no date>   myserver.crl
! Note that the next line indicates that the format is PKCS12.
     3   -rw-         1499             <no date>   myserver.p12
```

Example: PEM-Formatted Archive

The following sample output shows that autoarchiving has been configured in PEM file format. The archive consists of the CA certificate and the CA private key. To restore the certificate server using the backup, you would have to import the PEM-formatted CA certificate and CA key individually.

Note In addition to the CA certificate and CA key archive files, you should also back up the serial file (.ser) and the CRL file (.crl) regularly. The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

```
Device# more nvram:mycs.pem

-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDgyNzAyMzI0NloXDTA3MDgyNzAyMzI0NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1lZpKP4nGDJHgPkpYSkix7lD
```

```
nr23aMlZ9Kz5oo/qTBxeZ8mujpjYcZ0T8AZvoOiCuDnYMl796ZwpkMgjz1aZZbL+
BtuVvllsEOfhC+u/Ol/vxfGG5xpshoz/F5J3xdg5ZZuWWuIDAUYu9+QbI5feuG04
Z/BiPIb4AmGTP4B2MM0CAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUKi/cuK6wkz+ZswVtb06vUJboEeEwHQYDVR0O
BBYEFCov3LiusJM/mbMFbW9Or1CW6BHhMA0GCSqGSIb3DQEBBAUAA4GBAKLOmoE2
4+NeOKEXMCXG1jcohK7O2HrkFfl/vpK0+q92PTnMUFhxLOqI8pWIq5CCgC7heace
OrTv2zcUAoH4rzx3Rc2USIxkDokWWQMLujsMm/SLIeHit0G5uj//GCcbgK20MAW6
ymf7+TmblSFljWzstoUXC2hLnsJIMq/KffaD
-----END CERTIFICATE-----

!The private key is protected by the password that is
configured in "database archive pem password pwd" or that
is entered when you are prompted for the password.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,106CE91FFD0A075E

zyiFC8rKv8Cs+IKsQG2QpsVpvDBHqZqBSM4D528bvZv7jzr6WuHj8E6zO+6G8R/A
zjsfTALo+e+ZDg7KMzbryHARvjskbqFdOMLlVIYBhCeSElKsskWB6chOuyPHJInW
JwC5YzZdZwOqcyLBP/xOYXcvjzzNfPAXZzN12VR8vWDNq/kHT+3Lplc8hY++ABMI
M+C9FB3dpNZzu5O1BZCJg46bqbkulaCCmScIDaVt0zDFZwWTSufiemmNxZBG4xS8
t5t+FEhmSfv8DAmwg4f/KVRFTm10phUArcLxQO38Al0W5YHHORdACnuzVUvHgco7
VT4XUTjO7qMhmJgFNWy1pu49fbdS2NnOn5IoiyAq5lk1KUPrz/WABWiCvLMylGnZ
kyMCWoaMtgS/vdx74BBCj09yRZJnLMlIi6SDofjCNTDHfmFEVg4LsSWCd4lP9OP8
0MqhP1D5VIx6PbMNwkWW12lpBbCCdesFRGHjZD2dOu96kHD7ItErx34CC8W04aG4
b7DLktUu6WNV6M8g3CAqJiC0V8ATlp+kvdHZVkXovgND5IU0OJpsj0HhGzKAGpOY
KTGTUekUboISjVVkI6efp1vO6temVL3Txg3KGhzWMJGrq1snghE0KnV8tkddv/9N
d/t1l+we9mrccTq50WNDnkEi/cwHI/0PKXg+NDNH3k3QGpAprsqGQmMPdqc5ut0P
86i4cF9078QwWg4Tpay3uqNH1Zz6UN0tcarVVNmDupFESUxYw10qJrrEYVRadu74
rKAU4Ey4xkAftB2kuqvr21Av/L+jne4kkGIoZYdB+p/M98pQRgkYyg==
-----END RSA PRIVATE KEY-----
```

Example: Restoring a Certificate Server from Certificate Server Backup Files

The following example shows that restoration is from a PKCS12 archive and that the database URL is NVRAM (the default).

```
Device# copy tftp://192.0.2.71/backup.ser nvram:mycs.ser

Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)

Device# copy tftp://192.0.2.71/backup.crl nvram:mycs.crl

Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)

Device# configure terminal
Device(config)# crypto pki import mycs pkcs12 tftp://192.0.2.71/backup.p12 cisco123

Source filename [backup.p12]?
CRYPTO_PKI: Imported PKCS12 file successfully.

Device(config)# crypto pki server mycs
! fill in any certificate server configuration here

Device(cs-server)# no shutdown
% Certificate Server enabled.

Device(cs-server)# end
Device# show crypto pki server
```

```
Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 34885330 B13EAD45 196DA461 B43E813F
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 01:49:13 GMT Aug 28 2007
    CRL NextUpdate timer: 01:49:16 GMT Sep 4 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following example shows that restoration is from a PEM archive and that the database URL is flash:

```
Device# copy tftp://192.0.2.71/backup.ser flash:mycs.ser

Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://192.0.2.71/backup.crl flash:mycs.crl
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)
Device# configure terminal

! Because CA cert has Digital Signature usage, you need to import using the "usage-keys"
keyword

Device(config)# crypto ca import mycs pem usage-keys terminal cisco123
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----
% Enter PEM-formatted encrypted private SIGNATURE key.
% End with "quit" on a line by itself.
! Paste the CA private key from .pem archive.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,5053DC842B04612A

1CnlF5Pqvd0zp2NLZ7iosxzTy6nDeXPpNyJpxB5q+V29IuY8Apb6TlJCU7YrsEB/
nBTK7K76DCeGPlLpcuyEI171QmkQJ2gA0QhC0LrRo09WrINVH+b4So/y7nffZkVb
p2yDpZwqoJ8cmRH94Tie0YmzBtEh6ayOud11z53qbrsCnfSEwszt1xrW1MKrFZrk
/fTy6loHzGFzl3BDj4r5gBecExwcPp74ldHO+Ld4Nc9egG8BYkeBCsZZOQNVhXLN
I0tODOs6hP915zb6OrZFYv0NK6grTBO9D8hjNZ3U79jJzsSP7UNzIYHNTzRJiAyu
i56Oy/iHvkCSNUIK6zeIJQnW4bSoM1BqrbVPwHU6QaXUqlNzZ8SDtw7ZRZ/rHuiD
RTJMPbKquAzeuBss1132OaAUJRStjPXgyZTUbc+cWb6zATNws2yijPDTR6sRHoQL
47wHMr2Yj80VZGgkCSLAkL88ACz9TfUiVFhtfl6xMC2yuFl+WRk1XfF5VtWe5Zer
3Fn1DcBmlF7O86XUkiSHP4EV0cI6n5ZMzVLx0XAUtdAl1gD94y1V+6p9PcQHLyQA
pGRmj5IlSFw90aLafgCTbRbmC0ChIqHy91UFa1ub0130+yu7LsLGRlPmJ9NE61JR
bjRhlUXItRYWY7C4M3m/0wz6fmVQNSumJM08RHq6lUB3olzIgGIZlZkoaESrLG0p
qq2AENFemCPF0uhyVS2humMHjWuRr+jedfc/IMl7sLEgAdqCVCfV3RZVEaNXBud1
4QjkuTrwaTcRXVFbtrVioT/puyVUlpA7+k7w+F5TZwUV08mwvUEqDw==
```

```
-----END RSA PRIVATE KEY-----
quit
% Enter PEM-formatted SIGNATURE certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive again.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private ENCRYPTION key.
% End with "quit" on a line by itself.
! Because the CA cert only has Digital Signature usage, skip the encryption part.
quit
% PEM files import succeeded.
Device(config)# crypto pki server mycs
Device(cs-server)# database url flash:

! Fill in any certificate server configuration here.
Device(cs-server)# no shutdown

% Certificate Server enabled.
Device(cs-server)# end
Device# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: F04C2B75 E0243FBC 19806219 B1D77412
    Granting mode is: manual
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 21:02:55 GMT Sep 2 2007
    CRL NextUpdate timer: 21:02:58 GMT Sep 9 2004
    Current storage dir: flash:
    Database Level: Minimum - no cert data written to storage
```

Example: Subordinate Certificate Server

The following configuration and output is typical of what you might see after configuring a subordinate certificate server. Please be aware that “ms2” refers to a 2048-bit RSA key that was generated in an earlier step.

```
Device(config)# crypto pki trustpoint sub
Device(ca-trustpoint)# enrollment url http://192.0.2.6
Device(ca-trustpoint)# rsa keypair ms2 2048
Device(ca-trustpoint)# exit
Device(config)# crypto pki server sub
Device(cs-server)# mode sub-cs
Device(ca-server)# no shutdown

%Some server settings cannot be changed after CA certificate generation.
```

```
% Please enter a passphrase to protect the private key
% or type Return to exit
Password:
Jan  6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan  6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan  6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically [OK]
Jan  6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2
% Do you accept this certificate? [yes/no]:
Jan  6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan  6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan  6 22:32:46.798: CRYPTO_CS: created a new serial file.
Jan  6 22:32:46.798: CRYPTO_CS: authenticating the CA 'sub'y
Trustpoint CA certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
Router (cs-server)#
Jan  6 22:33:30.562: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan  6 22:33:32.450: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan  6 22:33:32.454: CRYPTO_CS: exit FSM: new state check failed
Jan  6 22:33:32.454: CRYPTO_CS: cs config has been locked
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint MD5: CED89E5F 53B9C60E
> AA123413 CDDAD964
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 70787C76 ACD7E67F
7D2C8B23 98CB10E7 718E84B1
% Exporting Certificate Server signing certificate and keys...
Jan  6 22:34:53.839: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 22:34:53.843: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 22:34:53.843: CRYPTO_CS: starting enabling checks
Jan  6 22:34:53.843: CRYPTO_CS: nvram filesystem
Jan  6 22:34:53.883: CRYPTO_CS: found existing serial file.
Jan  6 22:34:53.907: CRYPTO_CS: old router cert flag 0x4
Jan  6 22:34:53.907: CRYPTO_CS: new router cert flag 0x44
Jan  6 22:34:56.511: CRYPTO_CS: DB version
Jan  6 22:34:56.511: CRYPTO_CS: last issued serial number is 0x1
Jan  6 22:34:56.551: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 22:34:56.551: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 22:34:56.603: CRYPTO_CS: SCEP server started
Jan  6 22:34:56.603: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:34:56.603: CRYPTO_CS: cs config has been locked
Jan  6 22:35:02.359: CRYPTO_CS: enter FSM: input state enabled, input signal time set
Jan  6 22:35:02.359: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:35:02.359: CRYPTO_CS: cs config has been locked
```

Example: Root Certificate Server Differentiation

When issuing certificates, the root certificate server (or parent subordinate certificate server) differentiates the certificate request from “Sub CA,” “RA,” and peer requests, as shown in the following sample output:

```
Device# crypto pki server server1 info req

Enrollment Request Database:
RA certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
Subordinate CS certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
1     pending        CB9977AD8A73B146D3221749999B0F66 hostname=host-subcs.company.com
RA certificate requests:
```

```
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
Router certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
```

Example: Show Output for a Subordinate Certificate Server

The following **show crypto pki server command**output indicates that a subordinate certificate server has been configured:

```
Device# show crypto pki server

Certificate Server sub:
  Status: enabled
  Server's configuration is locked  (enter "shut" to unlock it)
  Issuer name: CN=sub
  CA cert fingerprint: 11B586EE 3B354F33 14A25DDD 7BD39187
  Server configured in subordinate server mode
  Upper CA cert fingerprint: 328ACC02 52B25DB8 22F8F104 B6055B5B
  Granting mode is: manual
  Last certificate issued serial number: 0x1
  CA certificate expiration timer: 22:33:44 GMT Jan 6 2006
  CRL NextUpdate timer: 22:33:29 GMT Jan 13 2005
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

Example: RA Mode Certificate Server

The following output is typical of what you might see after having configured an RA mode certificate server:

```
Device-ra(config)# crypto pki trustpoint myra
Device-ra(ca-trustpoint)# enrollment url http://192.0.2.17
! Include "cn=ioscs RA" or "ou=ioscs RA" in the subject-name.
Device-ra(ca-trustpoint)# subject-name cn=myra, ou=ioscs RA, o=company, c=us
Device-ra(ca-trustpoint)# exit
Device-ra(config)# crypto pki server myra
Device-ra(cs-server)# mode ra
Device-ra(cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
Certificate has the following attributes:
Fingerprint MD5: 32661452 0DDA3CE5 8723B469 09AB9E85
Fingerprint SHA1: 9785BBCD 6C67D27C C950E8D0 718C7A14 C0FE9C38
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Ready to request the CA certificate.
%Some server settings cannot be changed after the CA certificate has been requested.
Are you sure you want to do this? [yes/no]: yes
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: cn=myra, ou=ioscs RA, o=company, c=us
% The subject name in the certificate will include: Router-ra.company.com
% Include the router serial number in the subject name? [yes/no]: no
```

```
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto pki certificate' command will also show the fingerprint.
% Enrollment in progress...
Device-ra (cs-server)#

Sep 15 22:32:40.197: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 82B41A76 AF4EC87D
AAF093CD 07747D3A
Sep 15 22:32:40.201: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 897CDF40 C6563EAA
0FED05F7 0115FD3A 4FFC5231
Sep 15 22:34:00.366: %PKI-6-CERTRET: Certificate received from Certificate Authority

Device-ra(cs-server)# end
Device-ra# show crypto pki server

Certificate Server myra:
    Status: enabled
    Issuer name: CN=myra
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server is running in RA mode
    Server configured in RA mode
    RA cert fingerprint: C65F5724 0E63B3CC BE7AE016 BE0D34FE
    Granting mode is: manual
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following output shows the enrollment request database of the issuing certificate server after the RA has been enabled:

Note The RA certificate request is recognized by the issuing certificate server because "ou=ioscs RA" is listed in the subject name.

```
Device-ca# crypto pki server mycs info request

Enrollment Request Database:
Subordinate CA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
! The request is identified as RA certificate request.
RA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
12     pending    88F547A407FA0C90F97CDE8900A30CB0
hostname=Router-ra.company.com,cn=myra,ou=ioscs RA,o=company,c=us
Router certificates requests:
ReqID   State     Fingerprint                      SubjectName
--------------------------------------------------------------
! Issue the RA certificate.
Device-ca# crypto pki server mycs grant 12
```

The following output shows that the issuing certificate server is configured to issue a certificate automatically if the request comes from an RA:

```
Device-ca(config)# crypto pki server mycs
Device-ca(cs-server)# grant ra-auto

% This will cause all certificate requests already authorized by known RAs to be automatically
```

```
 granted.
Are you sure you want to do this? [yes/no]: yes
Router-ca (cs-server)# end
Device-ca# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server will issue certificate for requests from the RA.
    Granting mode is: auto for RA-authorized requests, manual otherwise
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 22:29:37 GMT Sep 15 2007
    CRL NextUpdate timer: 22:29:39 GMT Sep 22 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following example shows the configuration of “myra”, an RA server, configured to support automatic rollover from “myca”, the CA. After the RA server is configured, automatic granting of certificate reenrollment requests is enabled:

```
crypto pki trustpoint myra
 enrollment url
http://myca
 subject-name ou=iosca RA
 rsakeypair myra
crypto pki server myra
 mode ra
 auto-rollover
crypto pki server mycs
 grant auto rollover ra-cert
 auto-rollover 25
```

Example: Enabling CA Certificate Rollover to Start Immediately

The following example shows how to enable automated CA certificate rollover on the server mycs with the **crypto pki server**command. The **show crypto pki server** command then shows the current state of the mycs server and that the rollover certificate is currently available for rollover.

```
Device(config)# crypto pki server mycs rollover

Jun 20 23:51:21.211:%PKI-4-NOSHADOWAUTOSAVE:Configuration was
modified.  Issue "write memory" to save new IOS CA certificate
! The config has not been automatically saved because the config has been changed.
Device# show crypto pki server

Certificate Server mycs:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
    CA cert fingerprint:E7A5FABA 5D7AA26C F2A9F7B3 03CE229A
    Granting mode is:manual
    Last certificate issued serial number:0x2
    CA certificate expiration timer:00:49:26 PDT Jun 20 2008
    CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
    Rollover status:available for rollover
    ! Rollover certificate is available for rollover.
    Rollover CA certificate fingerprint:9BD7A443 00A6DD74 E4D9ED5F B7931BE0
```

```
Rollover CA certificate expiration time:00:49:26 PDT Jun 20 2011
Auto-Rollover configured, overlap period 25 days
```

Where to Go Next

After the certificate server is successfully running, you can either begin enrolling clients through manual mechanisms (as explained in the module "*Configuring Certificate Enrollment for a PKI*") or begin configuring SDP, which is a web-based enrollment interface, (as explained in the module "*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*.")

Additional References for Configuring and Managing a Certificate Server for PKI Deployment

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
PKI and security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
USB Token RSA Operations: Using the RSA keys on a USB token for initial autoenrollment	*Configuring Certificate Enrollment for a PKI*
USB Token RSA Operations: Benefits of using USB tokens	*Storing PKI Credentials*
Certificate server client certificate enrollment, autoenrollment, and automatic rollover	*Configuring Certificate Enrollment for a PKI*
Setting up and logging into a USB token	*Storing PKI Credentials*
Web-based certificate enrollment	*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*
RSA keys in PEM formatted files	*Deploying RSA Keys Within a PKI*
Choosing a certificate revocation mechanism	*Configuring Authorization and Revocation of Certificates in a PKI*

Related Topic	Document Title
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring and Managing a Certificate Server for PKI Deployment

CHAPTER 115

Storing PKI Credentials

Public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates can be stored in a specific location on the router, such as NVRAM and flash memory or on a USB eTtoken 64 KB smart card. USB tokens provide secure configuration distribution, RSA operations such as on-token key generation, signing, and authentication, and the storage of Virtual Private Network (VPN) credentials for deployment.

Prerequisites for Storing PKI Credentials

Prerequisites for Specifying a Local Certificate Storage Location

Before you can specify the local certificate storage location, your system should meet the following requirements:

- A Cisco IOS Release 12.4(2)T PKI-enabled image or a later image
- A platform that supports storing PKI credentials as separate files
- A configuration that contains at least one certificate
- An accessible local file system

Prerequisites for Specifying USB Token Storage for PKI Credentials

Before you can use a USB token, your system should meet the following requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, a Cisco 3800 series router, or a Cisco 7200VXR NPE-G2 platform
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms

- A Cisco supported USB token (Safenet/Aladdin eToken PRO 32 KB or 64 KB)
- A k9 image

Restrictions for Storing PKI Credentials

Restrictions for Specifying a Local Certificate Storage Location

When storing certificates to a local storage location, the following restrictions are applicable:

- Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
- A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.

Restrictions for Specifying USB Token Storage

When using a USB token to store PKI data, the following restrictions are applicable:

- USB token support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- You cannot boot an image from a USB token. (However, you can boot a configuration from a USB token.)
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports.

Information About Storing PKI Credentials

Storing Certificates to a Local Storage Location

Certificates are stored to NVRAM by default; however, some routers do not have the required amount of NVRAM to successfully store certificates.

All Cisco platforms support NVRAM and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token.

During run time, you can specify what active local storage device you would like to use to store certificates.

PKI Credentials and USB Tokens

To use a secure USB token on your router, you should understand the following concepts:

How a USB Token Works

A smart card is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A USB token is a smart card with a USB interface. The token can securely store any type of file within its available storage space (32 KB). Configuration files that are stored on the USB token can be encrypted

and accessed only via a user PIN. The device does not load the configuration file unless the proper PIN has been configured for secure deployment of device configuration files.

After you plug the USB token into the device, you must log into the USB token; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts (default: 15 attempts) before future logins are refused. For more information on accessing and configuring the USB token, see the section “Logging Into and Setting Up the USB Token."

After you have successfully logged into the USB token, you can copy files from the device on to the USB token via the **copy** command. USB token RSA keys and associated IPsec tunnels remain available until the device is reloaded. To specify the length of time before the keys are removed and the IPsec tunnels are torn down, issue the **crypto pki token removal timeout** command. The default timeout is zero, which causes the RSA keys to be removed automatically after the eToken is removed from the device. The default appears in the running configuration as:

```
crypto pki token default removal timeout 0
```

The table below highlights the capabilities of the USB token.

Table 157: Functionality Highlights for USB Tokens

Function	USB Token
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and device configurations from the USB token to the device.
Storage Size	32 KB or 64 KB
File Types	• Typically used to store digital certificates, preshared keys, and device configurations for IPsec VPNs. • USB tokens cannot store Cisco IOS images.
Security	• Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format.
Boot Configurations	• The device can use the configuration stored in the USB token during boot time. • The device can use the secondary configuration stored in the USB token during boot time. (A secondary configuration allows users to load their IPsec configuration.)

Benefits of USB Tokens

USB token support on a Cisco router provides the following application benefits:

Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

A USB token can use smart card technology to store a digital certificate and configuration for IPsec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPsec tunnel. (Because a router can initiate multiple IPsec tunnels, the USB token can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

PIN Configuration for Secure File Deployment

A USB token can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

The USB token can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, the USB token can store a bootstrap configuration that the router can use to boot from after the USB token has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

RSA Operations

A USB token may be used as a cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.

General-purpose, special-usage, encryption, or signature RSA key pairs with a modulus of 2048 bits or less may be generated from credentials located on your token storage device. Private keys are not distributed and remain on the token by default, however you may configure the private key storage location.

Keys that reside on a USB token are saved to persistent token storage when they are generated. Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from non-token storage locations when the **write memory** or a similar command is issued.)

Remote Device Configuration and Provisioning in a Secure Device Provisioning (SDP) Environment

SDP may be used to configure a USB token. The configured USB token may be transported to provision a device at a remote location. That is, a USB token may be used to transfer cryptographic information from one network device to another remote network device providing a solution for a staged USB token deployment.

For information about using USB tokens with SDP, see document titles in the “Additional References” section.

How to Configure PKI Storage

Specifying a Local Storage Location for Certificates

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki certificate storage** *location-name*
4. **exit**
5. **copy** *source-url destination-url*
6. **show crypto pki certificates storage**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki certificate storage** *location-name* **Example:** `Device(config)# crypto pki certificate storage flash:/certs`	Specifies the local storage location for certificates.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 5	**copy** *source-url destination-url* **Example:** `Device# copy system:running-config nvram:startup-config`	(Optional) Saves the running configuration to the startup configuration. **Note** Settings will only take effect when the running configuration is saved to the startup configuration.
Step 6	**show crypto pki certificates storage** **Example:** `Device# show crypto pki certificates storage`	(Optional) Displays the current setting for the PKI certificate storage location.

Example

The following is sample output from the **show crypto pki certificates storage** command, which shows that the certificates are stored in the certs subdirectory of disk0:

```
Device# show crypto pki certificates storage
Certificates will be stored in disk0:/certs/
```

Setting Up and Using USB Tokens on Cisco Devices

Storing the Configuration on a USB Token

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot config** *usbtoken[0-9]:filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**boot config** *usbtoken[0-9]:filename* **Example:** `Device(config)# boot config usbtoken0:file`	Specifies that the startup configuration file is stored in a secure USB token.

Logging Into and Setting Up the USB Token

How RSA Keys are Used with a USB Token

- RSA keys are loaded after the USB token is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted USB token. Regenerated keys should be stored in the same location where the original RSA key was generated.

Configuring the Device for Manual Login

Unlike automatic login, manual login requires that the user know the actual USB token PIN.

Note Either the manual or automatic login is required.

Manual login can be used when storing a PIN on the device is not desirable. Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the device is obtained from the local supplier or drop-shipped to the remote site. Manual login can be executed with or without privileges, and it creates files and RSA keys on the USB token available to the Cisco IOS software. If a secondary configuration

file is configured, it is executed only with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the USB token to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the device configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the USB token can provide. The USB token can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]
3. **show usbtoken** *0-9*:*filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* [**admin**] **login** [*pin*] **Example:** `Device# crypto pki token usbtoken0 admin login 5678`	Manually logs into the USB token. If the **admin** keyword is not specified initially you can re-enter the **crypto pki token** command again with this keyword option.
Step 3	**show usbtoken** *0-9*:*filename* **Example:** `Device# show usbtoken0:usbfile`	(Optional) Verifies whether the USB token has been logged on to the device.

What to Do Next

After you have logged into the USB token, it is available for use.

- To further configure the USB token, see the “Configuring the USB Token” section.
- To perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the “Setting Administrative Functions on the USB Token” section.

Configuring the USB Token

After you have set up automatic login, you may perform this task to further configure the USB token.

PINs and Passphrases

For additional PIN security with automatic login, you may encrypt your PIN stored in NVRAM and set up a passphrase for your USB token. Establishing a passphrase allows you to keep your PIN secure; another user needs only to know the passphrase, not the PIN.

When the USB token is inserted into the device, the passphrase is needed to decrypt the PIN. Once the PIN is decrypted, the device can then use the PIN to log in to the USB token.

Note The user needs a privilege level of 1 to log in.

Unlocking and Locking the USB Token

The USB token itself can be locked (encrypted) or unlocked (decrypted).

Unlocking the USB token allows it to be used. Once unlocked, Cisco IOS software treats the token as if it were automatically logged in. Any keys on the USB token are loaded, and if a secondary configuration file is on the token, it is executed with full user privileges (privilege level 15) independent of the privilege level of the logged-in user.

Locking the token, unlike logging out of the token, deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if configured.

Secondary Configuration and Unconfiguration Files

Configuration files that exist on a USB token are called secondary configuration files. If you create and configure a secondary configuration file, it is executed after the token is logged in. The existence of a secondary configuration file is determined by the presence of a secondary configuration file option in the Cisco IOS configuration stored in NVRAM. When the token is removed or logged out and the removal timer expires, a separate secondary unconfiguration file is processed to remove all secondary configuration elements from the running configuration. Secondary configuration and secondary unconfiguration files are executed at privilege level 15 and are not dependent on the level of the user logged in.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* **unlock** [*pin*]
3. **configure terminal**
4. **crypto pki token** *token-name* **encrypted-user-pin** [**write**]
5. **crypto pki token** *token-name* **secondary unconfig** *file*
6. **exit**
7. **crypto pki token** *token-name* **lock** [*pin*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**crypto pki token** *token-name* **unlock** [*pin*] **Example:** `Device# crypto pki token mytoken unlock mypin`	(Optional) Allows the token to be used if the USB token has been locked. Once unlocked, Cisco IOS software treats the token as if it has been automatically logged in. Any keys on the token are loaded and if a secondary configuration file exists, it is executed.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**crypto pki token** *token-name* **encrypted-user-pin** [**write**] **Example:** `Device(config)# crypto pki token mytoken encrypted-user-pin write`	(Optional) Encrypts the stored PIN in NVRAM.
Step 5	**crypto pki token** *token-name* **secondary unconfig** *file* **Example:** `Device(config)# crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg`	(Optional) Specifies the secondary configuration file and its location.
Step 6	**exit** **Example:** `Device(config)# exit`	Enters privileged EXEC mode.
Step 7	**crypto pki token** *token-name* **lock** [*pin*] **Example:** `Device# crypto pki token mytoken lock mypin`	(Optional) Deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if it exists.

Examples

The following example shows both the configuration and encryption of a user PIN and then the device reloading and the user PIN being unlocked:

```
! Configuring the user PIN

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# crypto pki token usbtoken0: userpin

Enter password: mypassword
```

```
! Encrypt the user PIN

Device(config)# crypto pki token usbtoken0: encrypted-user-pin

Enter passphrase: mypassphrase

Device(config)# exit

Device#

Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console

Device# show running config

crypto pki token usbtoken0 user-pin *encrypted*

! Reloading the router.

Device> enable

Password:

! Decrypting the user pin.

Device# crypto pki token usbtoken0: unlock

Token eToken is usbtoken0

Enter passphrase: mypassphrase

Token login to usbtoken0(eToken) successful

Device#

Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken

Login Successful
```

The following example shows a how a secondary unconfiguration file might be used to remove secondary configuration elements from the running configuration. For example, a secondary configuration file might be used to set up a PKI trustpoint. A corresponding unconfiguration file, named mysecondaryunconfigfile.cfg, might contain this command line:

```
no crypto pki trustpoint token-tp
```

If the token were removed and the following commands executed, the trustpoint and associated certificates would be removed from the device's running configuration:

```
 Device# configure terminal
Device(config)# no crypto pki token mytoken secondary unconfig mysecondaryunconfigfile.cfg
```

What to Do Next

After you have logged into and configured the USB token, it is available for use. If you want to perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the "Setting Administrative Functions on the USB Token" section.

Setting Administrative Functions on the USB Token

Perform this task to change default settings, such as the user PIN, the maximum number of failed attempts on the USB token, or the credential storage location.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* **admin**] **change-pin** [*pin*]
3. **crypto pki token** *token-name device-name***: label** *token-label*
4. **configure terminal**
5. **crypto key storage** *device-name***:**
6. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *device-name***:**] [**redundancy**] [**on** *device-name*]**:**
7. **crypto key move rsa** *keylabel* [**non-exportable** | [**on** | **storage**]] *location*
8. **crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]
9. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
10. **exit**
11. **copy usbflash**[*0-9*]**:***filename destination-url*
12. **show usbtoken**[*0-9*]**:***filename*
13. **crypto pki token** *token-name* **logout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* **admin**] **change-pin** [*pin*] **Example:** `Device# crypto pki token usbtoken0 admin change-pin`	(Optional) Changes the user PIN number on the USB token. • If the PIN is not changed, the default PIN 1234567890 is used. **Note** After the PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	**crypto pki token** *token-name device-name***: label** *token-label* **Example:** `Device# crypto pki token mytoken usb0: label newlabel`	(Optional) Sets or changes the name of the USB token. • The value of the *token-label* argument may be up to 31 alphanumeric characters in length including dashes and underscores.

	Command or Action	Purpose
		Tip This command is useful when configuring multiple USB tokens for automatic login, secondary configuration files, or other token specific settings.
Step 4	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 5	**crypto key storage** *device-name***:** **Example:** `Device(config)# crypto key storage usbtoken0:`	(Optional) Sets the default RSA key storage location for newly created keys. **Note** Regardless of configuration settings, existing keys are stored on the device from where they were originally loaded.
Step 6	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *device-name***:**] [**redundancy**] [**on** *device-name*]**:** **Example:** `Device(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:`	(Optional) Generates the RSA key pair for the certificate server. • The **storage** keyword specifies the key storage location. • When specifying a label name by specifying the *key-label* argument, you must use the same name for the label that you plan to use for the certificate server (through the **crypto pki server** *cs-label* command). If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the device, is used. If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the **no shutdown** command, then use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate and the private key. • By default, the modulus size of a CA key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range for a modulus size of a CA key is from 350 to 4096 bits. • The **on** keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:). **Note** Keys created on a USB token must be 2048 bits or less.

	Command or Action	Purpose
Step 7	**crypto key move rsa** *keylabel* [**non-exportable** | [**on** | **storage**]] *location* **Example:** `Device(config)# crypto key move rsa keypairname non-exportable on token`	(Optional) Moves existing Cisco IOS credentials from the current storage location to the specified storage location. By default, the RSA key pair remains stored on the current device. Generating the key on the device and moving it to the token takes less than a minute. Generating a key on the token, using the **on** keyword could take five to ten minutes, and is dependent on hardware key generation routines available on the USB token. When an existing RSA key pair is generated in Cisco IOS, stored on a USB token, and used for an enrollment, it may be necessary to move those existing RSA key pairs to an alternate location for permanent storage. This command is useful when using SDP with USB tokens to deploy credentials.
Step 8	**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*] **Example:** `Device(config)# crypto pki token usbtoken0 removal timeout 60`	(Optional) Sets the time interval, in seconds, that the device waits before removing the RSA keys that are stored in the USB token after the USB token has been removed from the device. **Note** If this command is not issued, all RSA keys and IPsec tunnels associated with the USB token are torn down immediately after the USB token is removed from the device.
Step 9	**crypto pki token** {*token-name* | **default**} **max-retries** [*number*] **Example:** `Device(config)# crypto pki token usbtoken0 max-retries 20`	(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the USB token is denied. • By default, the value is set at 15.
Step 10	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 11	**copy usbflash**[*0-9*]**:***filename destination-url* **Example:** `Device# copy usbflash0:file1 nvram:`	Copies files from USB token to the device. • *destination-url*—See the **copy** command page documentation for a list of supported options.
Step 12	**show usbtoken**[*0-9*]**:***filename* **Example:** `Device# show usbtoken:usbfile`	(Optional) Displays information about the USB token. You can use this command to verify whether the USB token has been logged in to the device.

	Command or Action	Purpose
Step 13	**crypto pki token** *token-name* **logout** **Example:** `Device# crypto pki token usbtoken0 logout`	Logs the device out of the USB token. **Note** If you want to save any data to the USB token, you must log back into the token.

Troubleshooting USB Tokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB token:

Troubleshooting the USB Port Connection

Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module.
- The Cisco IOS image running on the router does not support a USB module.
- A hardware problem with the USB module itself.

Sample output from the **show file systems** command showing a USB token appears below. The USB module listing appears in the last line of the examples.

```
Device# show file systems
File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
      491512      486395     nvram     rw   nvram:
           -           -    opaque     wo   syslog:
           -           -    opaque     rw   xmodem:
           -           -    opaque     rw   ymodem:
           -           -   network     rw   rcp:
           -           -   network     rw   pram:
           -           -   network     rw   ftp:
           -           -   network     rw   http:
           -           -   network     rw   scp:
           -           -   network     rw   https:
           -           -    opaque     ro   cns:
    63158272    33037312  usbflash     rw   usbflash0:
       32768         858  usbtoken     rw   usbtoken1:
```

Determining if a USB Token is Supported by Cisco

Use the **show usb device** command to determine if a USB token is supported by Cisco. The following output from this command indicates whether or not the module is supported is bold in the sample output below:

```
Router# show usb device
```

```
Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA
    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

Determining USB Token Device Problems

Use the **show usb controllers** command to determine if there is a hardware problem with a USB flash module. If the **show usb controllers** command displays an error, the error indicates a hardware problem in the USB module.

You can also use the **show usb controllers** command to verify that copy operations onto a USB flash module are occurring successfully. Issuing the **show usb controllers** command after performing a file copy should display successful data transfers.

The following sample output for the **show usb controllers** command displays a working USB flash module:

```
Router# show usb controllers
Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
```

```
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF
Int Level:1
Transfer Completion Codes:
         Success              :920            CRC              :0
         Bit Stuff            :0              Stall            :0
         No Response          :0              Overrun          :0
         Underrun             :0              Other            :0
         Buffer Overrun       :0              Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2              Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0              Bulk Transfer   :0
         Isochronous Transfer :0              Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0              Bulk Transfer   :26
         Isochronous Transfer :0              Control Transfer:894
USBD Failures:
         Enumeration Failures :0              No Class Driver Found:0
         Power Budget Exceeded:0
USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3              Command Fail    :0
         Good Status Timed out:0              Device not Found:0
         Device Never Opened  :0              Drive Init Fail :0
         Illegal App Handle   :0              Bad API Command :0
         Invalid Unit Number  :0              Invalid Argument:0
         Application Overflow :0              Device in use   :0
         Control Pipe Stall   :0              Malloc Error    :0
         Device Stalled       :0              Bad Command Code:0
         Device Detached      :0              Unknown Error   :0
         Invalid Logic Unit Num:0
USB Aladdin Token Driver Counters:
         Token Inserted       :1              Token Removed   :0
         Send Insert Msg Fail :0              Response Txns   :434
         Dev Entry Add Fail   :0              Request Txns    :434
         Dev Entry Remove Fail:0              Request Txn Fail:0
         Response Txn Fail    :0              Command Txn Fail:0
         Txn Invalid Dev Handle:0
USB Flash File System Counters:
         Flash Disconnected   :0              Flash Connected :1
         Flash Device Fail    :0              Flash Ok        :1
         Flash startstop Fail :0              Flash FS Fail   :0
USB Secure Token File System Counters:
         Token Inserted       :1              Token Detached  :0
         Token FS success     :1              Token FS Fail   :0
         Token Max Inserted   :0              Create Talker Failures:0
```

```
Token Event          :0               Destroy Talker Failures:0
Watched Boolean Create Failures:0
```

Displaying USB Token Infomation

Use the **dir** command with the **filesystem** keyword option **usbtoken***0-9***:** to display all files, directories, and their permission strings on the USB token.

The following sample output displays directory information for the USB token:

```
Device# dir usbtoken1:
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices to which the device is aware:

```
Device# dir all-filesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles
No space information available
Directory of flash:/
    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T
129880064 bytes total (99753984 bytes free)
Directory of nvram:/
  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data
491512 bytes total (486395 bytes free)
Directory of usbflash0:/
    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

Configuration Examples for PKI Storage

Example: Storing Certificates to a Specific Local Storage Location

The following configuration example shows how to store certificates to the certs subdirectory. The certs subdirectory does not exist and is automatically created.

```
Router# dir nvram:
 114  -rw-        4687                  <no date>  startup-config
 115  ----        5545                  <no date>  private-config
 116  -rw-        4687                  <no date>  underlying-config
   1  ----          34                  <no date>  persistent-data
   3  -rw-         707                  <no date>  ioscaroot#7401CA.cer
   9  -rw-         863                  <no date>  msca-root#826E.cer
  10  -rw-         759                  <no date>  msca-root#1BA8CA.cer
  11  -rw-         863                  <no date>  msca-root#75B8.cer
  24  -rw-        1149                  <no date>  storagename#6500CA.cer
  26  -rw-         863                  <no date>  msca-root#83EE.cer
129016 bytes total (92108 bytes free)
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
  14  -rw-         707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
  15  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
  16  -rw-         759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
  17  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
  18  -rw-        1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
  19  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)
! The certificate files are now on disk0/certs:
```

Example: Logging Into a USB Token and Saving RSA Keys to the USB Token

The following configuration example shows to how log in to the USB token, generate RSA keys, and store the RSA keys on the USB token:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
```

```
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
 ! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **show crypto key mypubkey rsa** command displays stored credentials after they are successfully loaded from the USB token. Credentials that are stored on the USB token are in the protected area. When storing the credentials on the USB token, the files are stored in a directory called /keystore. However, the key files are hidden from the command-line interface (CLI).

```
Router#
show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

Additional References

Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	*Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide*
eToken and USB flash data sheet	*USB eToken and USB Flash Features Support*
RSA keys	Deploying RSA Keys Within a PKI
File management (loading, copying, and rebooting files)	*Cisco Configuration Fundamentals Configuration Guide* on Cisco.com
USB Token RSA Operations: Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" feature document. See the "Generating a Certificate Server RSA Key Pair" section, the "Configuring a Certificate Server Trustpoint" section, and related examples.
USB Token RSA Operations: Using USB tokens for RSA operations upon initial autoenrollment	See the "Configuring Certificate Enrollment or Autoenrollment" section of the "Configuring Certificate Enrollment for a PKI " feature document.
SDP setup, configuration and use with USB tokens	See the feature information section for the feature names on using SDP and USB tokens to deploy PKI credentials in the "Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" feature document.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Storing PKI Credentials

Table 158: Feature Information for Storing PKI Credentials

Feature Name	Releases	Feature Information
Certificate -- Storage Location Specification		This feature allows you to specify the storage location of local certificates for platforms that support storing certificates as separate files. All Cisco platforms support NVRAM, which is the default location, and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token. The following commands were introduced by this feature: **crypto pki certificate storage**, **show crypto pki certificates storage**.
RSA 4096-bit Key Generation in Software Crypto Engine Support	15.1(1)T	The range value for the **modulus** keyword value for the **crypto key generate rsa** command is extended from 360 to 2048 bits to 360 to 4096 bits.

CHAPTER 116

Source Interface Selection for Outgoing Traffic with Certificate Authority

The Source Interface Selection for Outgoing Traffic with Certificate Authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

Information About Source Interface Selection for Outgoing Traffic with Certificate Authority

Certificates That Identify an Entity

Certificates can be used to identify an entity. A trusted server, known as the certification authority (CA), issues the certificate to the entity after determining the identity of the entity. A router that is running Cisco IOS XE software obtains its certificate by making a network connection to the CA. Using the Simple Certificate Enrollment Protocol (SCEP), the router transmits its certificate request to the CA and receives the granted certificate. The router obtains the certificate of the CA in the same manner using SCEP. When validating a certificate from a remote device, the router may again contact the CA or a Lightweight Directory Access Protocol (LDAP) or HTTP server to determine whether the certificate of the remote device has been revoked. (This process is known as checking the certificate revocation list [CRL].)

Note Depending on your Cisco IOS release, LDAP is supported.

In some configurations, the router may make the outgoing TCP connection using an interface that does not have a valid or routable IP address. The user must specify that the address of a different interface be used as the source IP address for the outgoing connection. Cable modems are a specific example of this requirement because the outgoing cable interface (the RF interface) usually does not have a routable address. However, the user interface (usually FastEthernet) does have a valid IP address.

Source Interface for Outgoing TCP Connections Associated with a Trustpoint

The **crypto pki trustpoint** command is used to specify a trustpoint. The **source interface**command is used along with the **crypto pki trustpoint**command to specify the address of the interface that is to be used as the source address for all outgoing TCP connections associated with that trustpoint.

Note If the interface address is not specified using the **source interface**command, the address of the outgoing interface is used.

How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority

Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint

Perform this task to configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **source interface** *interface-address*
6. **interface** *type slot / port*
7. **description** *string*
8. **ip address** *ip-address mask*
9. **interface** *type slot / port*
10. **description** *string*
11. **ip address** *ip-address mask*
12. **crypto map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint ms-ca`	Declares the Certificate Authority (CA) that your router should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://yourname:80/certsrv/mscep/mscep.dll`	Specifies the enrollment parameters of your CA.
Step 5	**source interface** *interface-address* **Example:** `Router (ca-trustpoint)# interface fastethernet1/0`	Interface to be used as the source address for all outgoing TCP connections associated with that trustpoint.
Step 6	**interface** *type slot / port* **Example:** `Router (ca-trustpoint)# interface fastethernet1/0`	Configures an interface type and enters interface configuration mode.
Step 7	**description** *string* **Example:** `Router (config-if)# description inside interface`	Adds a description to an interface configuration.
Step 8	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 9	**interface** *type slot / port* **Example:** `Router (config-if)# interface fastethernet1/0`	Configures an interface type.

	Command or Action	Purpose
Step 10	**description** *string* **Example:** `Router (config-if)# description outside interface 10.1.1.205 255.255.255.0`	Adds a description to an interface configuration.
Step 11	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 10.2.2.205 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 12	**crypto map** *map-name* **Example:** `Router (config-if)# crypto map mymap`	Applies a previously defined crypto map set to an interface.

Troubleshooting Tips

Ensure that the interface specified in the command has a valid address. Attempt to ping the router using the address of the specified interface from another device (possibly the HTTP or LDAP server that is serving the CRL). You can do the same thing by using a traceroute to the router from the external device.

You can also test connectivity between the router and the CA or LDAP server by using Cisco IOS XE command-line interface (CLI). Enter the **ping ip**command and respond to the prompts. If you answer "yes" to the "Extended commands [n]:" prompt, you will be able to specify the source address or interface.

In addition, you can use Cisco IOS XE CLI to input a traceroute command. If you enter the **traceroute ip** command (in EXEC mode), you will be prompted for the destination and source address. You should specify the CA or LDAP server as the destination and the address of the interface that you specified in the "source interface" as the source address.

Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority

Source Interface Selection for Outgoing Traffic with Certificate Authority Example

In the following example, the router is located in a branch office. The router uses IP Security (IPSec) to communicate with the main office. FastEthernet 1 is the "outside" interface that connects to the Internet Service Provider (ISP). FastEthernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the router must send its IP datagrams out interface FastEthernet 1 (address 10.2.2.205) using the IPSec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the router is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the **source interface** command tells the router to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This scenario is configured using the **source interface** command and the interface addresses as described above.

```
crypto pki trustpoint ms-ca
 enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
 source interface fastethernet0
!
interface fastethernet 0
 description inside interface
 ip address 10.1.1.1 255.255.255.0
!
interface fastethernet 1
 description outside interface
 ip address 10.2.2.205 255.255.255.0
 crypto map main-office
```

Additional References

The following sections provide references related to the Source Interface Selection for Outgoing Traffic with Certificate Authority feature.

Related Documents

Related Topic	Document Title
Configuring IPSec and certification authority	Security for VPNs with IPsec
IPSec and certification authority commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature.	-

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	-

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

Table 159: Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

Feature Name	Releases	Feature Information
Source Interface Selection for Outgoing Traffic with Certificate Authority.	Cisco IOS XE Release 2.1	This feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured. The following command was introduced: **source interface**.

Glossary

authenticate--Toprove the identity of an entity using the certificate of an identity and a secret that the identity poses (usually the private key corresponding to the public key in the certificate).

CA --Certificate Authority. A CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CA authentication --The user manually approves a certificate from a root CA. Usually a fingerprint of the certificate is presented to the user, and the user is asked to accept the certificate based on the fingerprint. The

certificate of a root CA is signed by itself (self-signed) so that it cannot be automatically authenticated using the normal certificate verification process.

CRL --certificate revocation list. A CRL is a data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire.

enrollment --A router receives its certificate via the enrollment process. The router generates a request for a certificate in a specific format (known as PKCS #10). The request is transmitted to a CA, which grants the request and generates a certificate encoded in the same format as the request. The router receives the granted certificate and stores it in an internal database for use during normal operations.

certificate --A data structure defined in International Organization for Standardization (ISO) standard X.509 to associate an entity (machine or human) with the public key of that entity. The certificate contains specific fields, including the name of the entity. The certificate is normally issued by a CA on behalf of the entity. In this case the router will act as its own CA. Common fields within a certificate include the distinguished name (DN) of the entity, the DN of the authority issuing the certificate, and the public key of the entity.

LDAP --Lightweight Directory Access Protocol. A LDAP is a protocol that provides access for management and browser applications that provide read-and-write interactive access to the X.500 directory.

CHAPTER 117

PKI Trustpool Management

The PKI Trustpool Management feature is used to authenticate sessions, such as HTTPS, that occur between devices by using commonly recognized trusted agents called certificate authorities (CAs).

Trustpool certificates are well-known CA certificates with which you can establish trust. IOS PKI has both built-in CAs and also has an option to download trustpool bundle. Built-in CA certificates are used to verify PKCS7 signature of downloaded trustpool bundle. You can download the trustpool bundle if signature verification fails. You can delete Built-in trustpool certificates. Trustpool certificates are used by applications such as SSLVPN, PnP, Smart License, MacSec and so on.

This feature, which is enabled by default, is used to create a scheme to provision, store, and manage a pool of certificates from known CAs in a way similar to the services a browser provides for securing sessions.

Note A new root certificate is included in the built-in certificates for Cisco Plug and Play application.

Note Effective with Cisco IOS XE Denali 16.3, the way PKI Trustpools are managed have changed. If you are planning to upgrade to this release, please review the changes to the feature captured below as part of *PKI Trustpool Enhancements* section.

Prerequisites for PKI Trustpool Management

The use of certificates requires that a crypto subsystem is included in the Cisco IOS software image.

Restrictions for PKI Trustpool Management

Device certificates that use CA certificates cannot be enrolled in a PKI trustpool.

You can download only a Cisco signed PKCS7 certificate through the trustpool URL.

Information About PKI Trustpool Management

CA Certificate Storage in a PKI Trustpool

The router uses a built-in CA certificate bundle that is contained in a special certificate store called a PKI trustpool, which is updated automatically from Cisco. This PKI trustpool is known by Cisco and other vendors. A CA certificate bundle can be in the following formats:

- X.509 certificates in Distinguished Encoding Rules (DER) binary format enveloped within a public-key cryptographic message syntax standard 7 (pkcs7), which is used to sign and encrypt messages under a PKI. An X.509 certificate is a PKI and Privilege Management Infrastructure (PMI) standard that specifies, among other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.
- A file containing concatenated X.509 certificates in Privacy Enhanced Mail (PEM) format with PEM headers.

Note Flash can also be used as the storage location for the bundles instead of NVRAM.

PKI Trustpool Updating

The PKI trustpool is treated as a single entity that needs to be updated when the following conditions occur:

- A certificate in the PKI trustpool is due to expire or has been reissued.
- The published CA certificate bundle contains additional trusted certificates that are needed by a given application.
- The configuration has been corrupted.

Note A built-in certificate in the PKI trustpool cannot be physically replaced. However, a built-in certificate is rendered inactive after an update if its X.509 subject-name attribute matches the certificate in the CA certificate bundle.

The PKI trustpool can be updated automatically or manually. The PKI trustpool may be used by certficate validation depending upon the application using it. See the "Manually Updating Certificates in the PKI Trustpool" and "Configuring Optional PKI Trustpool Policy Parameters" sections for more information.

Note When auto-update is enabled, all the existing downloaded trustpool certificates (excluding inbuilt trustpool certificates) will be deleted regardless of the import method.

The PKI trustpool timer matches the CA certificate with the earliest expiration time. If the timer is running and a bundle location is not configured and not explicitly disabled, syslog warnings are issued to alert the administrator that the PKI trustpool policy option is not set.

Automatic PKI trustpool updates use the configured URL.

When the PKI trustpool expires, the policy is read, the bundle is loaded, and the PKI trustpool is replaced. If the automatic PKI trustpool update encounters problems when initiating, then the following schedule is used to initiate the update until the download is successful: 20 days, 15 days, 10 days, 5 days, 4 days, 3 days, 2 days, 1 day, and then once every hour.

CA Handling in Both PKI Trustpool and Trustpoint

There may be circumstances where a CA resides in both PKI trustpool and trustpoint; for example, a trustpoint uses a CA and a CA bundle is downloaded later with the same CA inside. In this scenario, the CA in the trustpoint and the policy of this trustpoint is considered before the CA in the PKI trustpool or PKI trustpool policy to ensure that any current behavior is not altered when the PKI Trustpool Management feature is implemented on the router.

PKI Trustpool Enhancements

In releases earlier than Cisco IOS XE Denali 16.3, the trustpool consists of built-in certificates deployed with every Cisco box and downloaded CA certificates from published bundles. The downloaded certificates are saved in NVRAM, by default. The certificates from the downloaded trustpool bundle would be extracted and stored in the running configuration which was inefficient and utilized too much space.

From Cisco IOS XE Denali 16.3, the PKI trustpool enhancements stores the bundles in the same downloaded bundle format as one file in the storage location (default is NVRAM) instead of individual certificates like in the previous releases. This helps in saving storage memory as the file is in compressed format. Also, the certificates are not displayed individually in the running configuration. On every reboot the bundles are read from the storage location and individual certificates are installed in the database.

This feature removes the current downloaded certificates from the running configuration. The **crypto pki certificate pool** will not have the DER format certificates because these certificates are incompatible with the old NVRAM file and the new images. During upgrade, the trustpool certificates in DER format are lost and the bundles must be reinstalled again in the storage. This is indicated by a syslog during reboot in case of old NVRAM files. The **show crypto pki trustpool** command indicates that the configuration has been removed. Before you upgrade, use the **show crypto pki trustpool** command to verify that the certificates are available.

The following steps must be followed before upgrading to Cisco IOS XE Denali 16.3 :

- Remove the downloaded trustpool certificates using the **crypto pki trustpool clean** command
- Use the **write memory** command
- Reboot the device
- Download the trustpool bundles using the **crypto pki trustpool import url** command

If you are using trustpool certificates to log into SSH, then you need to follow additional steps to transfer that specific certificate from bundle to a trustpoint. See *Example: Using PKI Trustpool for SSH Connection During Upgrade* for more information.

Note From Cisco IOS XE Gibraltar 16.10 release onwards, when you configure the **match crlsign** command under trustpoint, the crlsign will be crosss checked while validating.

How to Configure PKI Trustpool Management

Manually Updating Certificates in the PKI Trustpool

The PKI Trustpool Management feature is enabled by default and uses the built-in CA certificate bundle in the PKI trustpool, which receives automatic updates from Cisco. Perform this task to manually update certificates in the PKI trustpool if they are not current, are corrupt, or if certain certificates need to be updated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpool import clean** [**terminal** | **url** *url*]
4. **crypto pki trustpool import** {**terminal**} {**url** *url* | **ca-bundle**} {**vrf** *vrf-name* | **source interface** *interface-name*}
5. **exit**
6. **show crypto pki trustpool**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpool import clean** [**terminal**	**url** *url*] **Example:** `Device(config)# crypto pki trustpool import clean`	(Optional) Manually removes all downloaded PKI CA certificates. • The **clean** keyword specifies the removal of the downloaded PKI trustpool certificates before the new certificates are downloaded.

	Command or Action	Purpose
		• The **terminal** keyword removes the existing CA certificate bundle terminal setting. • The **url** keyword and *url* argument removes the existing URL file system setting.
Step 4	**crypto pki trustpool import** {**terminal**} {**url** *url* | **ca-bundle**} {**vrf** *vrf-name* | **source interface** *interface-name*} **Example:** `Device(config)# crypto pki trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b`	Manually imports (downloads) the CA certificate bundle into the PKI trustpool to update or replace the existing CA certificate bundle. • The **terminal** keyword specifies the importation of a CA certificate bundle through the terminal (cut-and-paste) in PEM format. • The **url** keyword with the *url* argument specifies the importation of a CA certificate bundle through a URL. This URL can be through a variety of URL file systems such as HTTP. See the *PKI Trustpool Updating* section for more information. In CA bundle, you can use the **crypto pki trustpool import** command to pass the traffic through global VRF. Also, the traffic will not divert through a VRF, when you configure the **crypto pki trustpool policy** command specifying the VRF and source interface.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 6	**show crypto pki trustpool** **Example:** `Device(config)# show crypto pki trustpool`	Displays the PKI trustpool certificates of the router in a verbose format.

Configuring Optional PKI Trustpool Policy Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpool policy**
4. **cabundle url** {*url* | **none**}
5. **chain-validation**
6. **crl** {**cache** {**delete-after** {*minutes* | **none**} | **query** *url*}
7. **default** *command-name*

8. **match certificate** *certificate-map-name* [**allow expired-certificate** | **override** {**cdp directory** *ldap-location* | **ocsp** {*number* **url** *url* | **trustpool** *name number* **url** *url*} | **sia** *number url*} | **skip** [**revocation-check** | **authorization-check**]]
9. **ocsp** {**disable-nonce** | **url** *url*}
10. **revocation-check** *method1* [*method2* [*method3*]]
11. **source interface** *name number*
12. **storage** *location*
13. **vrf** *vrf-name*
14. **show**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**crypto pki trustpool policy**
	Example:
	`Device(config)# crypto pki trustpool policy`
	`Device(ca-trustpool)#`
		Enters ca-trustpool configuration mode where commands can be accessed to configure CA PKI trustpool policy parameters. The trustpool policy only affects the crl retrieval process and has no effect on trustpool import process.

Step 4	**cabundle url** {*url*	**none**} **Example:** `Device(ca-trustpool)# cabundle url http://www.cisco.com/security/pki/crl/crca2048.crl`	Specifies the URL from which the PKI trustpool certificate authority CA certificate bundle is downloaded . • The *url* argument is the URL of the CA certificate bundle. • The **none** keyword specifies that autoupdates of the PKI trustpool CA are not permitted.
Step 5	**chain-validation** **Example:** `Device(ca-trustpool)# chain-validation`	Enables chain validation from the peer's certificate to the root CA certificate in the PKI trustpool. The default has validation stopping at the peer certificate's issuer.
Step 6	**crl** {**cache** {**delete-after** {*minutes*	**none**}	**query** *url*} **Example:** `Device(ca-trustpool)# crl query http://www.cisco.com/security/pki/crl/crca2048.crl`	Specifies the certificate revocation list (CRL) query and CRL cache options for the PKI trustpool. • The **cache** keyword specifies CRL cache options. • The **delete-after** keyword removes the CRL from the cache after a timeout.

	Command or Action	Purpose
		• The *minutes* argument is the number of minutes from 1 to 43,200 to wait before deleting the CRL from the cache. • The **none** keyword specifies that CRLs are not cached. • The **query** keyword with the *url* argument specifies the URL published by the CA server to query the CRL.
Step 7	**default** *command-name* **Example:** `Device(ca-trustpool)# default crl query http://www.cisco.com/security/pki/crl/crca2048.crl`	Resets the value of a ca-trustpool configuration subcommand to its default . • The *command-name* argument is the ca-trustpool configuration mode command with its applicable keywords.
Step 8	**match certificate** *certificate-map-name* [**allow expired-certificate** | **override** {**cdp directory** *ldap-location* | **ocsp** {*number* **url** *url* | **trustpool** *name number* **url** *url*} | **sia** *number url*} | **skip** [**revocation-check** | **authorization-check**]] **Example:** `match certificate mycert override ocsp 1 url http://ocspts.identrust.com`	Enables the use of certificate maps for the PKI trustpool. • The *certifcate-map-name* argument matches the certificate map name. • The optional **allow expired-certificate** keyword ignores expired certificates. **Note** If this keyword is not configured, the router does not ignore expired certificates. • The **override** keyword overrides the online certificate status protocol (OCSP) or SubjectInfoAccess (SIA) attribute fields in a certificate that is in the PKI trustpool. • The **cdp** keyword overrides the certificate distribution point (CDP) in a certificate. • The **directory** keyword and *ldap-location* specifies the CDP in either the http: or ldap: URL, or LDAP directory to override in the certificate. • The **ocsp** keyword and *number* argument and **url** keyword and *url* argument specifies the OCSP sequence number from 0 to 10000 and URL to override in the certificate. • The **trustpool** keyword and *name* and *number* arguments with the **url** keyword and *url* argument override the PKI trustpool for verifying the OCSP certificate by specifying the PKI trustpool name, sequence number, and URL.

	Command or Action	Purpose
		• The **sia** keyword and *number* and *url* arguments override the SIA URL in a certificate by specifying the SIA sequence number and URL. • The optional **skip revocation-check** keyword combination allows the PKI trustpool to enforce certificate revocation lists (CRLs) except for specific certificates. **Note** If this keyword combination is not configured, then the PKI trustpool enforces CRLs for all certificates. • The optional **skip authorization-check** keyword combination skips the authentication, authorization, and accounting (AAA) check of a certificate when public key infrastructure (PKI) integration with an AAA server is configured. **Note** If this keyword combination is not configured, and PKI integration with an AAA server is configured, then the AAA checking of a certificate is done.
Step 9	**ocsp** {**disable-nonce** | **url** *url*} **Example:** `Device(ca-trustpool)# ocsp url http://ocspts.identrust.com`	Specifies OCSP settings for the PKI trustpool. • The **disable-nonce** keyword disables the OCSP Nonce extension. • The **url** keyword and *url* argument specify the OCSP server URL to override (if one exists) in the Authority Info Access (AIA) extension of the certificate. All certificates associated with a configured PKI trustpool are checked by the OCSP server at the specified HTTP URL. The URL can be a hostname, IPv4 address, or an IPv6 address.
Step 10	**revocation-check** *method1* [*method2* [*method3*]] **Example:** `Device(ca-trustpool)# revocation-check ocsp crl none`	Disables revocation checking when the PKI trustpool policy is being used. The *method* argument is used by the router to check the revocation status of the certificate. Available keywords are as follows: • The **crl** keyword performs certificate checking by a certificate revocation list (CRL). This is the default behavior. • The **none** keyword does not require a certificate checking. • The **ocsp** keyword performs certificate checking by an online certificate status protocol (OCSP) server.

	Command or Action	Purpose
		If a second and third method are specified, each method is used only if the previous method returns an error, such as a server being down.
Step 11	**source interface** *name number* **Example:** `Device(ca-trustpool)# source interface tunnel 1`	Specifies the source interface to be used for CRL retrieval, OCSP status, or the downloading of a CA certificate bundle for the PKI trustpool . • The *name* and *number*arguments are for the interface type and number used as the source address for the PKI trustpool.
Step 12	**storage** *location* **Example:** `Device(ca-trustpool)# storage storage disk0:crca2048.crl`	Specifies a file system location where PKI trustpool certificates are stored on the router. • The *location* is the file system location where the PKI trustpool certificates are stored. The types of file system locations are **disk0:**, **disk1:**, **nvram:**, **unix:**, or a named file system.
Step 13	**vrf** *vrf-name* **Example:** `Device(ca-trustpool)# vrf myvrf`	Specifies the VPN routing and forwarding (VRF) instance to be used for enrolment, CRL retrieval, and OCSP status.

Step 14	**show**
	Example:
	`Device(ca-trustpool)# show`
	
	`Chain validation will stop at the first CA certificate in the pool`
	`  Trustpool CA certificates will expire 12:58:31 PST Apr 5 2012`
	`  Trustpool policy revocation order:  crl`
	`  Certficate matching is disabled`
	`  Policy Overrides:`
		Displays the PKI trustpool policy of the router.

Configuration examples for PKI Trustpool Management

Example: Configuring PKI Trustpool Management

The following **show crypto pki trustpool** command output displays the certificates in PKI trustpool:

Note The command output in this example is abridged because it is verbose.

```
Device# show crypto pki trustpool
```

```
CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 00D01E474000000111C38A964400000002
  Certificate Usage: Signature
  Issuer:
    cn=DST Root CA X3
    o=Digital Signature Trust Co.
  Subject:
    cn=Cisco SSCA
    o=Cisco Systems
  CRL Distribution Points:
    http://crl.identrust.com/DSTROOTCAX3.crl
  Validity Date:
    start date: 12:58:31 PST Apr 5 2007
    end   date: 12:58:31 PST Apr 5 2012

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 6A6967B3000000000003
  Certificate Usage: Signature
  Issuer:
    cn=Cisco Root CA 2048
    o=Cisco Systems
  Subject:
    cn=Cisco Manufacturing CA
    o=Cisco Systems
  CRL Distribution Points:
    http://www.cisco.com/security/pki/crl/crca2048.crl
  Validity Date:
    start date: 14:16:01 PST Jun 10 2005
    end   date: 12:25:42 PST May 14 2029
```

The following **show crypto pki trustpool verbose** command output displays the certificates in PKI trustpool:

```
Device# show crypto pki trustpool verbose

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=Licensing Root - DEV
    o=Cisco
  Subject:
    cn=Licensing Root - DEV
    o=Cisco
  Validity Date:
    start date: 03:25:43 IST Apr 25 2013
    end   date: 03:25:43 IST Apr 25 2033
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: 432CBFA0 32D2983A 8A56A319 FD28C6F9
  Fingerprint SHA1: 6341FCAF 19CE9FEE 961D92A5 D47390B5 2DD6D94D
  X509v3 extensions:
```

```
    X509v3 Key Usage: 6000000
      Key Cert Sign
      CRL Signature
    X509v3 Subject Key ID: 43214521 B5FB217A 1A4D1BB7 0236E664 CBEC8B65
    X509v3 Basic Constraints:
        CA: TRUE
    Authority Info Access:
  Associated Trustpoints: Trustpool
  Trustpool: Built-In
```

Example: Using PKI Trustpool for SSH Connection During Upgrade

Before upgrading to Cisco IOS XE Denali 16.3, copy the certificate from trustpool to a new trustpoint.

```
Device # show run | sec pool
crypto pki trustpool policy
 revocation-check none
 source interface GigabitEthernet0/0/0
crypto pki certificate pool
 certificate ca 01
  308204FA 308202E2 A0030201 02020101 300D0609 2A864886 F70D0101 0C050030
  0E310C30 0A060355 04031303 61626330 1E170D31 36303730 35303435 3935335A
  170D3136 30373035 30353535 35335A30 0E310C30 0A060355 04031303 61626330
  82022230 0D06092A 864886F7 0D010101 05000382 020F0030 82020A02 82020100
  C78AA144 8EC1D18A 4EECC3E8 81450CC7 A85A4C57 AF59E584 5C1EA888 6EF70DA8
  33327D93 E1F6CED7 32BB4FCF 693F60E0 37000225 40F6F9C5 0462C4AD 899E5BDD
  ED779180 D6C75E1B FBE97D42 E2A7B35D DDC18C4D 4CCDE401 68F67A6D E40FD744
  904EE49F 40820640 C6E0B072 510BC40E A0883F6C E8DF5128 EFF3B5F4 B31E5C16
  217652FF AFC30EBF 593CB19C 56C0E793 2814D504 0E079E0C 8E9E856A BCADB19C
  F2376994 A0A040C1 7BC1E88F CF80F218 9C48B4D9 F84ED5C0 79827BD1 32448478
  8F1F82F2 C91A9479 692B6456 C53CF937 777D0C31 1B8A1F5E 24B33553 047C2448
  855CF974 DFA21665 8AD8A0E5 81ED8068 81688997 FF05118C 93A59CA0 7FD594F6
  B7B1898C 272E089A 3392A2C4 22A22625 2BC1E16F 95B2FC15 207CCA49 378AD3A6
  0C574197 C5E94D8C E6736271 CE0BA9AB ACB380E3 A8084243 4E038DD1 8E86E206
  E2269290 F1AFB29A D28CFB3A 5ABADE4A 21A59728 7174E7A3 2FF59C90 E6100C6E
  E2E8CB4C 91BD574D 57B5E18A 78F9CE75 624C4A2E 1A6EFCC3 7D1BB20B 1CC79024
  CD2FBC4D 46BE1B7A 6EFD8F05 6FD84E91 51215E9B E5E952A4 6E2D1388 10075706
  7D6FAF9B 3F7F8994 F39B9B5D 0C7CD5BC 40738877 5D9985AC 5AB6363D 811BA440
  41A1639F 352F4F01 1994300A A4B85B75 01486CA0 4C4B3175 82038B26 BEFE1D2A
  4AC0D577 7784FACF A6877D68 5D73DD04 DC8D942B DE3FC9FE 4C1FF715 A2E7A5AB
  02030100 01A36330 61300F06 03551D13 0101FF04 05300301 01FF300E 0603551D
  0F0101FF 04040302 0186301F 0603551D 23041830 168014CA 195EDBF1 51753A92
  71342CA8 36DDABA9 63A93130 1D060355 1D0E0416 0414CA19 5EDBF151 753A9271
  342CA836 DDABA963 A931300D 06092A86 4886F70D 01010C05 00038202 0100553B
  FB77A348 C4447C40 BEB2DDFD 63C82441 3CBDC198 B5D5B1AB DF17C4E2 98AEAF2F
  CD570939 BCC116E0 33CFF471 E91EE308 8B29B5BD 11DFACF9 A3AC3135 8BE81B22
  ED205587 5DE04654 A051CC14 CA8D2A6E 81F924DA 001BB1C4 7F85F177 4E75D8EA
  797CCAEF 1502492D 17627CD1 E39E295B 44C55884 8E6DFF68 2129B222 18E3187D
  AB97B4A7 6F838E75 A8908566 AD9E6687 35B150DE 0C8C1B37 6F17FDAC 7A7C53A4
  434F5CF3 6EB71957 E65EC5D2 7685B05B A9D8C0D3 2DB8F97E E6B37E11 C9E26F4F
  BFB97745 83E1A214 461B0E49 0FFDEF21 A7CA5364 44416002 03A01F0C 2BC098D3
  B50A4071 AC4D2234 4E55C5D4 0FD9C308 63F2A8D4 24D34613 B73EAA1B B407D56F
  90EEF5C7 AE61C0D8 13FB493D 0E1C8F9B 1D2D6DEA 458CDE18 8753FF14 F8C75213
  35557FCC 50405056 D9790AF0 EAC21646 2D9AF88D 59C05434 45F21248 0BB72191
  74D951DD 9D23997E 1134611E 837137E6 C40C694E 7AB4A05F E8470E87 E0F6D924
  A69A98A8 5AA2B9B3 B7446883 94A7230D EE3C6EDA 4A348351 FC40C16D 6FDC91EC
  CEFF580B F7826DD1 1D1D07DB 17CA3298 8C510826 D2712E04 EB669909 3D8106EB
  5391A5BA 80B7E981 B41AAEB9 CE4A5236 20E30AE7 01D5FDB3 604C5505 0F8C96DC
```

```
  8F5CF569 5D90C1FB F5679221 B7B922C0 5F11C379 9EBA283C 45A209F7 132B8DA2
  EAF4751B 290A1CAC C3E7978B 760FB05A 185991FE 4884FA1A D3EEDD7C 63
  3B
        quit
```

Paste the certificate in config mode by creating a new trustpoint.

```
Device(config)#cry pki trust abc
Device(ca-trustpoint)#cry pki cert chain abc
Device(config-cert-chain)#certificate ca 01
```

Enter the certificate in hexadecimal representation

```
Device(config-pki-hexmode)#  308204FA 308202E2 A0030201 02020101 300D0609 2A864886 F70D0101
 0C050030
Device(config-pki-hexmode)#  0E310C30 0A060355 04031303 61626330 1E170D31 36303730 35303435
 3935335A
Device(config-pki-hexmode)#  170D3136 30373035 30353535 35335A30 0E310C30 0A060355 04031303
 61626330
Device(config-pki-hexmode)#  82022230 0D06092A 864886F7 0D010101 05000382 020F0030 82020A02
 82020100
Device(config-pki-hexmode)#  C78AA144 8EC1D18A 4EECC3E8 81450CC7 A85A4C57 AF59E584 5C1EA888
 6EF70DA8
Device(config-pki-hexmode)#  33327D93 E1F6CED7 32BB4FCF 693F60E0 37000225 40F6F9C5 0462C4AD
 899E5BDD
Device(config-pki-hexmode)#  ED779180 D6C75E1B FBE97D42 E2A7B35D DDC18C4D 4CCDE401 68F67A6D
 E40FD744
Device(config-pki-hexmode)#  904EE49F 40820640 C6E0B072 510BC40E A0883F6C E8DF5128 EFF3B5F4
 B31E5C16
Device(config-pki-hexmode)#  217652FF AFC30EBF 593CB19C 56C0E793 2814D504 0E079E0C 8E9E856A
 BCADB19C
Device(config-pki-hexmode)#  F2376994 A0A040C1 7BC1E88F CF80F218 9C48B4D9 F84ED5C0 79827BD1
 32448478
Device(config-pki-hexmode)#  8F1F82F2 C91A9479 692B6456 C53CF937 777D0C31 1B8A1F5E 24B33553
 047C2448
Device(config-pki-hexmode)#  855CF974 DFA21665 8AD8A0E5 81ED8068 81688997 FF05118C 93A59CA0
 7FD594F6
Device(config-pki-hexmode)#  B7B1898C 272E089A 3392A2C4 22A22625 2BC1E16F 95B2FC15 207CCA49
 378AD3A6
Device(config-pki-hexmode)#  0C574197 C5E94D8C E6736271 CE0BA9AB ACB380E3 A8084243 4E038DD1
 8E86E206
Device(config-pki-hexmode)#  E2269290 F1AFB29A D28CFB3A 5ABADE4A 21A59728 7174E7A3 2FF59C90
 E6100C6E
Device(config-pki-hexmode)#  E2E8CB4C 91BD574D 57B5E18A 78F9CE75 624C4A2E 1A6EFCC3 7D1BB20B
 1CC79024
Device(config-pki-hexmode)#  CD2FBC4D 46BE1B7A 6EFD8F05 6FD84E91 51215E9B E5E952A4 6E2D1388
 10075706
Device(config-pki-hexmode)#  7D6FAF9B 3F7F8994 F39B9B5D 0C7CD5BC 40738877 5D9985AC 5AB6363D
 811BA440
Device(config-pki-hexmode)#  41A1639F 352F4F01 1994300A A4B85B75 01486CA0 4C4B3175 82038B26
 BEFE1D2A
Device(config-pki-hexmode)#  4AC0D577 7784FACF A6877D68 5D73DD04 DC8D942B DE3FC9FE 4C1FF715
 A2E7A5AB
Device(config-pki-hexmode)#  02030100 01A36330 61300F06 03551D13 0101FF04 05300301 01FF300E
 0603551D
Device(config-pki-hexmode)#  0F0101FF 04040302 0186301F 0603551D 23041830 168014CA 195EDBF1
 51753A92
Device(config-pki-hexmode)#  71342CA8 36DDABA9 63A93130 1D060355 1D0E0416 0414CA19 5EDBF151
 753A9271
```

```
Device(config-pki-hexmode)#  342CA836 DDABA963 A931300D 06092A86 4886F70D 01010C05 00038202
 0100553B
Device(config-pki-hexmode)#  FB77A348 C4447C40 BEB2DDFD 63C82441 3CBDC198 B5D5B1AB DF17C4E2
 98AEAF2F
Device(config-pki-hexmode)#  CD570939 BCC116E0 33CFF471 E91EE308 8B29B5BD 11DFACF9 A3AC3135
 8BE81B22
Device(config-pki-hexmode)#  ED205587 5DE04654 A051CC14 CA8D2A6E 81F924DA 001BB1C4 7F85F177
 4E75D8EA
Device(config-pki-hexmode)#  797CCAEF 1502492D 17627CD1 E39E295B 44C55884 8E6DFF68 2129B222
 18E3187D
Device(config-pki-hexmode)#  AB97B4A7 6F838E75 A8908566 AD9E6687 35B150DE 0C8C1B37 6F17FDAC
 7A7C53A4
Device(config-pki-hexmode)#  434F5CF3 6EB71957 E65EC5D2 7685B05B A9D8C0D3 2DB8F97E E6B37E11
 C9E26F4F
Device(config-pki-hexmode)#  BFB97745 83E1A214 461B0E49 0FFDEF21 A7CA5364 44416002 03A01F0C
 2BC098D3
Device(config-pki-hexmode)#  B50A4071 AC4D2234 4E55C5D4 0FD9C308 63F2A8D4 24D34613 B73EAA1B
 B407D56F
Device(config-pki-hexmode)#  90EEF5C7 AE61C0D8 13FB493D 0E1C8F9B 1D2D6DEA 458CDE18 8753FF14
 F8C75213
Device(config-pki-hexmode)#  35557FCC 50405056 D9790AF0 EAC21646 2D9AF88D 59C05434 45F21248
 0BB72191
Device(config-pki-hexmode)#  74D951DD 9D23997E 1134611E 837137E6 C40C694E 7AB4A05F E8470E87
 E0F6D924
Device(config-pki-hexmode)#  A69A98A8 5AA2B9B3 B7446883 94A7230D EE3C6EDA 4A348351 FC40C16D
 6FDC91EC
Device(config-pki-hexmode)#  CEFF580B F7826DD1 1D1D07DB 17CA3298 8C510826 D2712E04 EB669909
 3D8106EB
Device(config-pki-hexmode)#  5391A5BA 80B7E981 B41AAEB9 CE4A5236 20E30AE7 01D5FDB3 604C5505
 0F8C96DC
Device(config-pki-hexmode)#  8F5CF569 5D90C1FB F5679221 B7B922C0 5F11C379 9EBA283C 45A209F7
 132B8DA2
Device(config-pki-hexmode)#  EAF4751B 290A1CAC C3E7978B 760FB05A 185991FE 4884FA1A D3EEDD7C
 63
Device(config-pki-hexmode)#  3B
Device(config-pki-hexmode)#        quit
```

Now you can upgrade to Cisco IOS XE Denali 16.3. The certificate from trustpool would disappear but would still stay in trustpoint. Install the certificate in trustpool after the upgrade.

Additional References for PKI Trustpool Management

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PKI Trustpool Management

Table 160: Feature Information for PKI Trustpool Management

Feature Name	Releases	Feature Information
PKI Trustpool Management		The following commands were introduced or modified: **cabundle url**, **chain-validation (ca-trustpool)**, **crypto pki trustpool import**, **crypto pki trustpool policy**, **crl**, **default (ca-trustpool)**, **match certificate (ca-trustpool)**, **ocsp**, **show (ca-trustpool)**, **show crypto pki trustpool**, **source interface (ca-trustpool)**, **storage**, **vrf (ca-trustpool)**, **show crypto pki trustpool built-in**, **crypto pki trustpool import clean ca-bundle**.

CHAPTER 118

PKI Split VRF in Trustpoint

The PKI Split VRF in Trustpoint feature allows you to configure a VPN Routing and Forwarding (VRF) for certificate enrollment and revocation.

Information About PKI Split VRF in Trustpoint

Overview of PKI Split VRF in Trustpoint

The PKI Split VRF in Trustpoint feature allows you to configure VPN Routing and Forwarding (VRF) for certificate enrollment and for certificate revocation list (CRL) checking. The VRF is configured in the enrollment profile using the **enrollment url** command under the **crypto pki profile enrollment** command to attach the enrollment profile to a trustpoint. You can configure the same VRF for enrollment and CRL or configure different VRFs. Based on the configuration (enrollment or revocation), the corresponding VRF is selected and Simple Certificate Enrollment Protocol (SCEP) request is sent via the respective VRF.

To configure enrollment and CRL via different routing paths, you must configure the enrollment url command using the **crypto pki profile enrollment** command. This configured VRF acts as an enrollment VRF and the enrollment request goes via that VRF. However, the CRL uses the global VRF configured in the trustpoint using the

If no VRF is configured in the **enrollment url** command, the enrollment takes global enrollment that is configured in the **crypto pki trustpoint** command.

How to Configure PKI Split VRF in Trustpoint

Configuring the Split VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki profile enrollment** *label*
4. **enrollment url** *url* [**vrf** *vrf-name*]
5. **exit**
6. **show crypto pki profile**
7. **show crypto pki trustpoint**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki profile enrollment** *label* **Example:** `Device(config)# crypto pki profile enrollment pki_profile`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label* —Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 4	**enrollment url** *url* [**vrf** *vrf-name*] **Example:** `Device(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe vrf vrf1`	Specifies the URL and the VPN Routing and Forwarding (VRF) of the CA server to which to send certificate enrollment requests via HTTP or TFTP.
Step 5	**exit** **Example:** `Device(ca-profile-enroll)# exit`	Exits ca-profile-enroll configuration mode. • Enter this command a second time to exit global configuration mode.
Step 6	**show crypto pki profile** **Example:** `Device# show crypto pki profile`	(Optional) Displays information about PKI profile.

	Command or Action	Purpose
Step 7	**show crypto pki trustpoint** **Example:** `Device# show crypto pki trustpoint`	(Optional) Displays information about PKI trustpoints.

Configuration Examples for PKI Split VRF in Trustpoint

Example: Configuring the PKI Split VRF in Trustpoint

Enrollment and Certificate Revocation List Via Same VRF

The following example shows how to configure the enrollment and certificate revocation list (CRL) via the same VRF:

```
crypto pki trustpoint trustpoint1
 enrollment url http://10.10.10.10:80
 vrf vrf1
 revocation-check crl
```

Enrollment and Certificate Revocation List Via Different VRF

The following example shows how to configure the enrollment and certificate revocation list (CRL) via different VRF:

```
crypto pki profile enrollment pki_profile
 enrollment url http://10.10.10.10:80 vrf vrf2

crypto pki trustpoint trustpoint1
 enrollment profile pki_profile
 vrf vrf1
 revocation-check crl
```

Additional References for PKI Split VRF in Trustpoint

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 161: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 119

EST Client Support

The EST Client Support feature allows you to enable EST (Enrolment Over Secure Transport) for all trustpoints while using SSL or TLS to secure transport.

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 162: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

Information About EST Client Support

Overview of EST Client Support

The EST Client Support feature allows you to use Enrollment over Secure Transport (EST) as a certificate management protocol for provisioning certificates. With the existing SCEP enrollment integrated within the PKI component, the addition of EST will introduce a new component that will use SSL or TLS to secure the transport. PKI will store all certificates.

To enable EST support, the EST client is required to authenticate the server during TLS connection establishment. For this authentication, the TLS server may require the client's credentials.

Prerequisites for EST Client Support

- Enable the **ip http authentication fore-close** command.

Restrictions for EST Client Support

- The EST client supports only TLS 1.2
- The certificate Attribute request is not supported.
- CA-Certificate rollover is not supported.
- Certificate-less TLS authentication is not supported.
- HTTP-based client authentication is not supported.

How to Configure EST Client Support

Configuring a Trustpoint to Use EST

Perform this task to configure a trustpoint to use EST (Enrolment Over Secure Transport) by enabling the user to use the enrollment profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki profile enrollment***label*
4. **method-est**
5. **enrollment url***url* [**vrf** *vrf name*]
6. **enrollment credential** *label*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki profile enrollment***label* **Example:** `Device(config)# crypto pki profile enrollment pki_profile`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 4	**method-est** **Example:** `Device(ca-profile-enroll)# method-est`	Enables enrollment profile to select usage of EST.
Step 5	**enrollment url***url* [**vrf** *vrf name*] **Example:** `Device(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe vrf vrf1`	Specifies that an enrollment profile is to be used for certificate enrollment. **Note** If the authentication URL is not specified, then the enrollment URL will be considered for authentication.
Step 6	**enrollment credential** *label* **Example:** `Device(ca-profile-enroll)# enrollment credential test_label`	Provides the trustpoint credentials currently available in the profile for TLS client authentication.
Step 7	**exit** **Example:** `Device(ca-profile-enroll)# exit`	Exits ca-profile-enroll configuration mode.

Verifying the EST Client Support Configaration

You can use the following show commands to verify EST Client Support configuration.

- **show crypto pki profile**
- **show crypto pki trustpoints estclient status**

Configuration Examples for EST Client Support

Configuring a Trustpoint to Use EST

The following example shows how to configure a trustpoint to use Enrollment over Secure Transport (EST):

```
crypto pki profile enrollment pki_profile
 method-est
```

```
 enrollment url http://www.example.com/BigCA/est/simpleenroll.dll
 enrollment credential test_label
```

Verifying EST Client Support

The following sample output from the **show crypto pki trustpoints estclient status** command verifies EST Client Support configuration.

```
Router# show crypto pki trustpoints estclient status
Trustpoint estclient:
  Issuing CA certificate configured:
    Subject Name:
     cn=estExampleCA
    Fingerprint MD5: B9D0403C 7D33F1AA F9957796 CA6E86AA
    Fingerprint SHA1: F3698C9C DCB2B5F2 A38EBCB4 1DBA6A90 9F877A5B
  Router Signature certificate configured:
    Subject Name:
     cn=estclientrouter
    Fingerprint MD5: B740849B 37016DB7 A6797CE4 D6140D27
    Fingerprint SHA1: F032B015 50BB5742 2619EFC6 F1F0B8B1 31D9906D
  State:
    Keys generated ..... Yes (Signature, non-exportable)
    Issuing CA authenticated ..... Yes
    Certificate request(s) ..... Yes
```

The following sample output from the **show crypto pki certificate estclient** command shows the status before re-enrollement and after re-enrollment.

```
BEFORE REENROLLMENT

Router# show crypto pki certificate estclient


Certificate
  Status: Available
  Certificate Serial Number (hex): 2603
  Certificate Usage: Signature
  Issuer:
    cn=estExampleCA
  Subject:
    Name: estclientrouter
    cn=estclientrouter
  CRL Distribution Points:
    http://example.com/crl.pem
  Validity Date:
    start date: 19:31:24 GMT Feb 8 2019
    end   date: 19:31:24 GMT Feb 8 2020
    renew date: 19:35:50 GMT Feb 8 2019
  Associated Trustpoints: estclient

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 00ACFCD09D3182CBEB
  Certificate Usage: General Purpose
  Issuer:
    cn=estExampleCA
  Subject:
    cn=estExampleCA
  Validity Date:
    start date: 09:40:47 GMT Mar 28 2018
    end   date: 09:40:47 GMT Mar 28 2019
```

```
  Associated Trustpoints: estclient ROOT


AFTER REENROLLMENT

show crypto pki certificates estclient
Certificate
  Status: Available
  Certificate Serial Number (hex): 4B
  Certificate Usage: Signature
  Issuer:
    cn=estExampleCA
  Subject:
    Name: estclientrouter
    cn=estclientrouter
  CRL Distribution Points:
    http://example.com/crl.pem
  Validity Date:
    start date: 07:34:05 GMT Feb 9 2019
    end   date: 07:34:05 GMT Feb 9 2020
    renew date: 19:38:35 GMT Feb 8 2019
  Associated Trustpoints: estclient

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 00E5EEC53E0FBD597D
  Certificate Usage: General Purpose
  Issuer:
    cn=estExampleCA
  Subject:
    cn=estExampleCA
  Validity Date:
    start date: 04:59:30 GMT Dec 20 2018
    end   date: 04:59:30 GMT Dec 20 2019
  Associated Trustpoints: estclient ROOT_SEC
```

Additional References for EST Client Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 7030	*Enrollment over Secure Transport*
RFC 2818	*HTTP Over TLS*
RFC 6125	*Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)*
RFC 2510	*Internet X.509 Public Key Infrastructure Certificate Management Protocols*
RFC 4210	*Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 120

OCSP Response Stapling

The OCSP Response Stapling feature allows you to check the validity of a peer's user or device credentials contained in a digital certificate using Online Certificate Status Protocol (OCSP).

Information About OCSP Response Stapling

Overview of OCSP Response Stapling

Online Certificate Status Protocol (OCSP) is a method to check certificate revocation when a peer has to retrieve this revocation information and then validate it to check the certificate revocation status. In this method, the certification revocation status is limited by the peer's ability to reach an OCSP responder through the cloud or by the certificate sender's performance in retrieving the certificate revocation-information.

OCSP response stapling supports a new method to fetch the OCSP response for a device’s own certificates. This feature allows the device to obtain its own certificate revocation information by contacting the OCSP server and then sending this result along with its certificates directly to the peer. As a result, the peer does not require to contact the OCSP responder.

How to Configure OCSP Response Stapling

Configuring PKI Client to Request EKU Attribute

Perform this task to configure OCSP (Online Certificate Status Protocol) response stapling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*

4. **ocsp url** *url*
5. **eku request** *attribute*
6. **match eku** *attribute*
7. **revocation-check** *method1* [*method2* [*method3*]]
8. **exit**
9. **exit**
10. **show cry pki counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **a.** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint msca`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 4	**ocsp url** *url*
	Example:
	`Device(ca-trustpoint)# ocsp url http://ocsp-server`
	Example:
	`Device(ca-trustpoint)# ocsp url http://10.10.10.1:80`
	Example:
	`Device(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80`
		The *url* argument specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL overrides the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. All certificates associated with a configured trustpoint are checked by the OCSP server. The URL can be a hostname, IPv4 address, or an IPv6 address.
		Note Make sure that the OCSP request url is configured with the **ocsp url** *url* command and not with an http-proxy server.

Step 5	**eku request** *attribute* **Example:** `Device(ca-trustpoint)# eku request ssh-client`	Requests to include specified eku *attribute* in the certificate. This request, when configured on the PKI client, will be sent to the CA server during enrollment. The *attribute* argument can be one of the following: • client-auth • code-signing • email-protection • ipsec-end-system

	Command or Action	Purpose
		• ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client
Step 6	**match eku** *attribute* **Example:** `Device(ca-trustpoint)# match eku client-auth`	Allows PKI to validate a peer certificate only if the specified attribute is present in the certificate else validation fails. The *attribute* argument can be one of the following: • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client
Step 7	**revocation-check** *method1* [*method2* [*method3*]] **Example:** `Device(ca-trustpoint)# revocation-check ocsp none`	(Optional) Checks the revocation status of a certificate. • crl --Certificate checking is performed by a CRL. This is the default option. • none --Certificate checking is ignored. • ocsp --Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 10	**show cry pki counters** **Example:** `Device# show cry pki counters`	(Optional) Displays the PKI counters of the device.

Configuring PKI Server to Include EKU Attributes

Perform this task to configure OCSP (Online Certificate Status Protocol) response stapling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*
5. **eku request** *attribute*
6. **exit**
7. **exit**
8. **show crypto pki counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **a.** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on your system.

	Command or Action	Purpose
Step 4	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server server-pki`	Defines a label for the certificate server and enters certificate server configuration mode. **Note** If you manually generated an RSA key pair, the *cs-label* argument must match the name of the key pair.
Step 5	**eku request** *attribute* **Example:** `Device(cs-server)# eku request ssh-server`	Requests to include specified eku *attribute* in the certificate. The *attribute* argument can be one of the following: • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client
Step 6	**exit** **Example:** `Device(cs-server)# exit`	Exits cs-server configuration mode and returns to global configuration mode.
Step 7	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show crypto pki counters** **Example:** `Device# show crypto pki counters`	(Optional) Displays the PKI counters of the device.

Example

The following is sample output from the **show crypto pki counters**.

```
Device# show crypto pki counters

PKI Sessions Started: 0
PKI Sessions Ended: 0
PKI Sessions Active: 0
Successful Validations: 0
Failed Validations: 0
Bypassed Validations: 0
Pending Validations: 0
CRLs checked: 0
CRL - fetch attempts: 0
CRL - failed attempts: 0
CRL - rejected busy fetching: 0
OCSP - fetch requests: 0
OCSP - received responses: 0
OCSP - failed attempts: 0
OCSP - staple requests: 0
AAA authorizations: 0
```

Additional References for OCSP Response Stapling

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 2560	*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
RFC 4806	*Online Certificate Status Protocol (OCSP) Extensions to IKEv2*
RFC 5280	*Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
RFC 6187	*X.509v3 Certificates for Secure Shell Authentication*
RFC 6066	*Transport Layer Security (TLS) Extensions: Extension Definitions*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 163: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 121

Configuring Route Processor Redundancy for PKI

Route Processor Redundancy provides an alternative to the High System Availability feature. HSA enables a system to reset and use a standby Route Switch Processor, if the active RSP fails. Using RPR, you can reduce unplanned downtime because RPR enables a quicker switchover between an active and standby RSP if the active RSP experiences a fatal error.

Route Processor Redundancy feature currently available on Cisco ASR platforms with dual RP support such as ASR 1006, ASR 1009. and ASR 1013.

Note Route Processor Redundancy supports trustpool import.

Prerequisites for Configuring Route Processor Redundancy

- You must use the same memory in both RSPs because the secondary RSP must be able to support the primary RSP during a failover.

Restrictions for Configuring Route Processor Redundancy

- Route Processor Redundancy feature only supports platforms with dual RP support.
- Route Processor Redundancy is supported only on routers that support dual RSPs.
- It is not recomended to configure RA (Registration Authority) as it is not validated.

How To Configure Route Processor Redundancy

Configuring Route Processor Redundancy SSO Mode

```
configure terminal
redundancy
 mode sso
 main-cpu
  standby console enable
exit
```

Verifying Route Processor Redundancy

```
show redundancy states
show crypto pki server
show crypto pki certificates tname
```

Route Processor Redundancy SSO Mode Confuguration Example

Example for server side configuration:

```
asr1k(config)#ip http server

asr1k(config)#crypto pki trustpoint ROOTCA

asr1k(ca-trustpoint)#hash sha512

asr1k(ca-trustpoint)#revocation-check none

asr1k(ca-trustpoint)#rsakeypair  ROOTCA 2048

asr1k(ca-trustpoint)#crypto pki server ROOTCA

asr1k(cs-server)#issuer-name CN=ROOTCA C=pki

asr1k(cs-server)#lifetime certificate 00 00 15

asr1k(cs-server)#lifetime ca-certificate 00 00 25

asr1k(cs-server)#lifetime crl 6

asr1k(cs-server)#serial-number 0x1

asr1k(cs-server)#auto-rollover 00 00 24

% The archive password is not configured. Rollover CA keys and certificates will not be
automatically archived.

asr1k(cs-server)#grant auto

asr1k(cs-server)#database url tftp://<ip>//

% Server database url was changed. You need to move the

% existing database to the new location.
```

```
asr1k(cs-server)#database url p12 tftp://<ip>//
asr1k(cs-server)#database level complete
asr1k(cs-server)#database archive pkcs12 password <pwd>
asr1k(cs-server)#end
```

Example for client side configuration:

```
crypto pki trustpoint client
 enrollment url http://<ip>:80
 usage ike
 subject-name CN=R1 C=pki
 revocation-check crl
 rsakeypair client 2048
 hash sha512
```

Route Processor Redundancy SSO Mode Verification Example

```
show redundancy states
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 48

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
    Manual Swact = enabled
 Communications = Up

   client count = 132
 client_notification_TMR = 30000 milliseconds
          RF debug mask = 0x0
```

```
show crypto pki server
Certificate Server ROOTCA:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=ROOTCA C=pki
    CA cert fingerprint: F2BF3707 D9F6F5F3 E0D111D8 A8486437
    Granting mode is: auto
    Last certificate issued serial number (hex): 2
    CA certificate expiration timer: 14:15:50 IST Mar 31 2019
    CRL NextUpdate timer: 14:15:50 IST Mar 31 2019
    Current primary storage dir: tftp://9.45.3.3//
    Current storage dir for .p12 files: tftp://9.45.3.3//
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 0 days
    Autorollover timer: 13:51:50 IST Mar 31 2019
    Redundancy configured.  This is active.
```

Note Server is enabled only on active RP and is in disabled state in standby mode.

```
show crypto pki certificates client
Certificate
  Status: Available
  Certificate Serial Number (hex): 03
  Certificate Usage: General Purpose
  Issuer:
    cn=ROOTCA C=pki
  Subject:
    Name: asr1k
    hostname=asr1k
    cn=R1 C=pki
  Validity Date:
```

```
    start date: 00:42:04 IST Mar 11 2019
    end   date: 01:02:04 IST Mar 11 2019
  Associated Trustpoints: client


CA Certificate
  Status: Available
  Certificate Serial Number (hex): 02
  Certificate Usage: Signature
  Issuer:
    cn=ROOTCA C=pki
  Subject:
    cn=ROOTCA C=pki
  Validity Date:
    start date: 00:40:34 IST Mar 11 2019
    end   date: 00:40:34 IST Mar 9 2020
  Associated Trustpoints: client
```

PART XII

Zone-Based Policy Firewalls

CHAPTER 122

Zone-Based Policy Firewalls

This module describes the Cisco unidirectional firewall policy between groups of interfaces known as zones. Prior to the release of the Cisco unidirectional firewall policy, Cisco firewalls were configured only as an inspect rule on interfaces. Traffic entering or leaving the configured interface was inspected based on the direction in which the inspect rule was applied.

Note Cisco IOS XE supports Virtual Fragmentation Reassembly (VFR) on zone-based firewall configuration. When you enable the firewall on an interface by adding the interface to a zone, VFR is configured automatically on the same interface.

Feature Information for Zone-Based Policy Firewalls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 164: Feature Information for Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Zone-Based Firewall Reclassification	Cisco IOS XE Bengaluru 17.6.1	The Zone-Based Firewall reclassification feature is introduced. This feature enforces changes, if any, to a policy configuration on the existing sessions.

Feature Name	Releases	Feature Information
Smart Licensing support for Zone-Based Firewall on ASR1000	Cisco IOS XE Denali 16.3.1	The following command was modified: **show license all**.
Out-of-Order Packet Handling in Zone-Based Policy Firewall	Cisco IOS XE Release 3.5S	The Out-of-Order Packet Handling feature allows OoO packets to pass through the router and reach their destination if a session does not require DPI. All Layer 4 traffic with OoO packets are allowed to pass through to their destination. However, if a session requires Layer 7 inspection, the OoO packets are still dropped.
IOS-XE ZBFW Interop with Crypto VPN	Cisco IOS XE Release 3.17S	The IOS-XE ZBFW Interop with Crypto VPN feature supports the enabling of zone-based firewall under FlexVPN DVTI.
Zone-Based Firewall Support of Multipath TCP	Cisco IOS XE Release 3.13S	Multipoint TCP seamlessly works with zone-based firewall Layer 4 inspection. Multipoint TCP does not work with application layer gateways (ALGs) and application inspection and control (AIC).
Firewall—NetMeeting Directory (LDAP) ALG Support	Cisco IOS XE Release 3.1S	LDAP is an application protocol that is used for querying and updating information stored on directory servers. The Firewall—Netmeeting (LDAP) Directory ALG Support feature enables Cisco firewalls to support Layer 4 LDAP inspection by default. The following command was introduced: **match protocol**.
Debuggability Enhancement in Zone-Based Firewall (Phase-II)	Cisco IOS XE Release 3.10S	The Debuggability Enhancement Zone-Based Firewall feature provides severity levels for debug logs.
Zone-Based Firewall—Default Zone	Cisco IOS Release 2.6	The Zone-Based Firewall— Default Zone feature introduces a default zone that enables a firewall policy to be configured on a zone pair that consist of a zone and a default zone. Any interface without explicit zone membership belongs to the default zone.
Zone-Based Policy Firewalls	Cisco IOS Release 2.1	The Zone-Based Policy Firewall feature provides a Cisco IOS XE software unidirectional firewall policy between groups of interfaces known as zones.

Information About Zone-Based Policy Firewalls

The following sections provide detailed information about zone-based policy firewalls.

Top-Level Class Maps and Policy Maps

Top-level class maps allow you to identify the traffic stream at a high level. This is accomplished by using the **match access-group** and **match protocol** commands. Top-level class maps are also referred to as Layer

3 and Layer 4 class maps. Top-level policy maps allow you to define high-level actions by using the **inspect**,**drop**, and**pass** commands. You can attach policy maps to a target (zone pair).

Note Only inspect type policies can be configured on a zone pair.

Overview of Zones

A zone is a group of interfaces having similar functions or features. They help you specify where a Cisco IOS XE firewall should be applied. For example, on a device, Gigabit Ethernet interface 0/0/0 and Gigabit Ethernet interface 0/0/1 may be connected to the local LAN. These two interfaces are similar because they represent the internal network. So, they can be grouped into a zone for firewall configurations.

By default, the traffic between the interfaces in the same zone is not subject to any policy, and passes freely. Firewall zones are used for security features.

Note Zones may not span interfaces in different VPN routing and forwarding (VRF) instances.

For Dynamic Multipoint VPN (DMVPN) tunnels, zone-based firewall inspects and only evaluates the inner packet. Once the inner packet is encapsulated in Generic Routing Encapsulation (GRE) and Encapsulating Security Payload (ESP) payloads, it is forwarded without further inspection. For incoming packets, ESP and GRE decapsulation takes place before ZBF evaluation. It is not required to configure any explicit rules for ESP and GRE traffic on self to outside or outside to self zone pairs.

Security Zones

A security zone is a group of interfaces to which a policy can be applied.

Grouping interfaces into zones involves two procedures:

- Creating a zone so that interfaces can be attached to it.
- Configuring an interface to be a member of a given zone.

By default, traffic flows between interfaces that are members of the same zone.

When an interface is a member of a security zone, all traffic (except traffic going to the device or initiated by the device) between that interface and an interface in a different zone is dropped by default. To permit traffic to and from a zone-member interface and another interface, you must make that zone part of a zone pair, and apply a policy to that zone pair. If the policy permits traffic through inspect or pass actions, traffic can flow through the interface.

The following are the basic rules to consider when setting up zones:

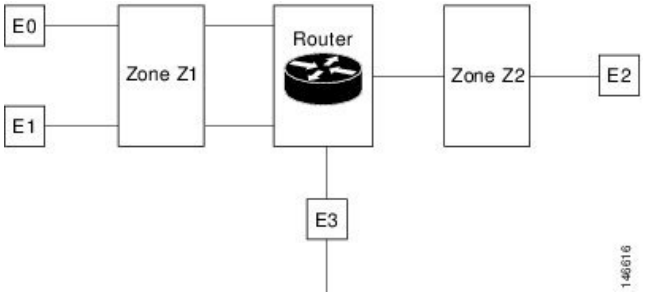
- Traffic from a zone interface to a nonzone interface, or from a nonzone interface to a zone interface is always dropped; unless default zones are enabled (default zone is a nonzone interface).
- Traffic between two zone interfaces is inspected if there is a zone pair relationship for each zone, and if there is a configured policy for that zone pair.
- By default, all traffic between two interfaces in the same zone is always allowed.

- A zone pair can be configured with a zone as both source and destination zones. An inspect policy can be configured on this zone pair to inspect, pass, or drop the traffic between the two zones.
- An interface can be a member of only one security zone.
- When an interface is a member of a security zone, all traffic to and from that interface is blocked unless you configure an explicit interzone policy on a zone pair involving that zone.
- For traffic to flow between all the interfaces in a device, these interfaces must be members of one security zone or another. It is not necessary for all the device interfaces to be members of security zones.
- All the interfaces associated with a zone must be contained in the same virtual routing and forwarding (VRF).

Figure 1 illustrates the following:

- Interfaces E0 and E1 are members of security zone Z1.
- Interface E2 is a member of security zone Z2.
- Interface E3 is not a member of any security zone.

Figure 51: Security Zone Restrictions



- The zone pair and policy are configured in the same zone. Traffic flows freely between interfaces E0 and E1 because they are members of the same security zone (Z1).
- If no policies are configured, traffic will not flow between any other interfaces, for example, E0 and E2, E1 and E2, E3 and E1, and E3 and E2.
- Traffic can flow between E0 or E1 and E2 only when an explicit policy permitting traffic is configured between zone Z1 and zone Z2.
- Traffic can never flow between E3 and E0, E1, or E2 unless default zones are enabled.

Note On the Cisco ASR 1000 Series Aggregation Services Routers, the firewall supports a maximum of 4000 zones.

Security Zone Firewall Policies

A class identifies a set of packets based on its contents. Normally, you define a class so that you can apply an action on the identified traffic that reflects a policy. A class is designated through class maps.

An action is a functionality that is typically associated with a traffic class. Firewall supports the following type of actions:

inspect: Once classified, firewall session is created in the connection table and the packet's content is examined.

pass: The packet is classified and the traffic is allowed to pass through the system without further inspection.

drop: The packet is classified and dropped.

To create security zone firewall policies, you must complete the following tasks:

- Define a match criterion (class map).
- Associate actions to the match criterion (policy map).
- Attach the policy map to a zone pair (service policy).

The **class-map** command creates a class map to be used for matching packets to a specified class. Packets that arrive at targets (such as the input interface, output interface, or zone pair), determined by how the **service-policy** command is configured, are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

The **policy-map** command creates or modifies a policy map that can be attached to one or more targets to specify a service policy. Use the **policy-map** command to specify the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Virtual Interfaces as Members of Security Zones

A virtual template interface is a logical interface configured with generic configuration information for a specific purpose or for a configuration common to specific users, plus device-dependent information. The template contains Cisco software interface commands that are applied to virtual access interfaces. To configure a virtual template interface, use the **interface virtual-template** command.

Zone member information is acquired from a RADIUS server, and the dynamically created interface is made a member of that zone. The **zone-member security** command adds the dynamic interface to the corresponding zone.

For more information on the Per Subscriber Firewall on LNS feature, see Release Notes for Cisco ASR 1000 Series Aggregation Services Routers for Cisco IOS XE Release 2.

Zone Pairs

A zone pair allows you to specify a unidirectional firewall policy between two security zones.

To define a zone pair, use the **zone-pair security** command. The direction of the traffic is specified by source and destination zones. The source and destination zones of a zone pair must be security zones.

You can select the default or self zone as either the source or the destination zone. The self zone is a system-defined zone that does not have any interfaces as members. A zone pair that includes the self zone, along with the associated policy, applies to traffic directed to the device or traffic generated by the device. It does not apply to traffic that is passing through the device.

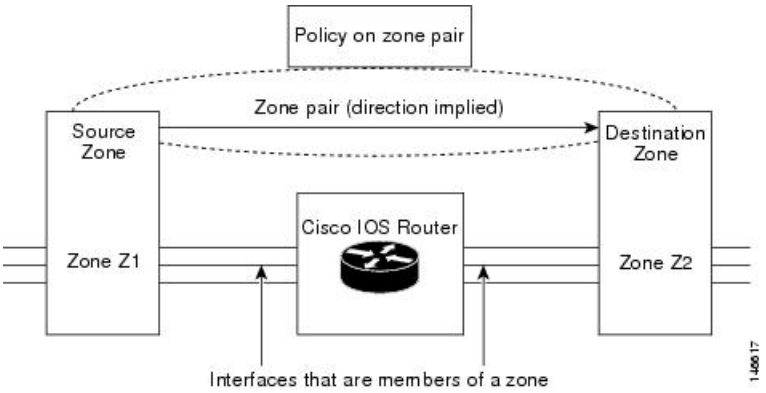
The default zone is applicable to interfaces where no security zone is associated. Default zones are not enabled by default. To enable default zones, use the **zone security default** configuration command.

Because the most common usage of firewall is applying them to traffic through a device, you need at least two zones. For traffic to and from the device, ZBF supports the concept of a self zone.

To permit traffic between zone member interfaces, you must configure a policy permitting (inspecting or passing) traffic between that zone and another zone. To attach a firewall policy map to the target zone pair, use the **service-policy type inspect** command.

The following figure shows the application of a firewall policy to traffic flowing from zone Z1 to zone Z2, which means that the ingress interface for the traffic is a member of zone Z1, and the egress interface is a member of zone Z2.

Figure 52: Zone Pairs



Since there are two zones, this might require policies for traffic going in both directions (from Z1 to Z2 and Z2 to Z1). If traffic is initiated from either direction, you must configure two zone pairs.

If a policy is not configured between zone pairs, traffic is dropped. However, it is not necessary to configure a zone pair and a service policy solely for the return traffic. By default, return traffic is not allowed. If a service policy inspects the traffic in the initiator direction and there is no zone pair and a service policy for the return traffic, the return traffic is inspected.

If a service policy passes the traffic in the forward direction and there is no zone pair and service policy for the return traffic, the return traffic is dropped. In both these cases, you need to configure a zone pair and a service policy to allow the return traffic. In figure 2, it is not mandatory that you configure a zone pair source and destination for allowing return traffic from Z2 to Z1. The service policy on the Z1 to Z2 zone pair takes care of it. For the pass action, a policy must exist for packets in each direction, and for the inspect action, a policy must exist for traffic from the initiator.

A zone-based firewall drops a packet if it is not explicitly allowed by a rule or policy in contrast to a legacy firewall, which permits a packet if it is not explicitly denied by a rule or policy by default.

A zone-based firewall behaves differently when handling intermittent Internet Control Message Protocol (ICMP) responses generated within a zone because of the traffic flowing between in-zones and out-zones.

A policy is not required for Internet Control Message Protocol (ICMP) error packets.

Note A policy is required for ICMP informational messages such as ICMP_ECHO (ping) for packets arriving from an initiator.

In a configuration where an explicit policy is configured for the self zone to go out of its zone and for the traffic moving between the in-zone and out-zone, if any informational ICMP packets, such as ICMP_EHCO_REQUEST are generated, then the zone-based firewall looks for an explicit permit rule for

the ICMP in the self zone to go out of its zone. An explicit inspect rule for the ICMP for the self zone to go out-zone may not help because no session is associated with the intermittent ICMP responses.

Zones and Inspection

Zone-based policy firewalls examine source and destination zones from the ingress and egress interfaces for a firewall policy. It is not necessary that all traffic flowing to or from an interface be inspected; you can designate that individual flows in a zone pair be inspected through your policy map that you apply across the zone pair. The policy map will contain class maps that specify individual flows. Traffic with the inspect action will create a connection in the firewall table and be subject to state checking. Traffic with the pass action will bypass the zone firewall completely, not creating any sessions. After a firewall connection is created, the packets are no longer classified. That is, if the policy map changes, the underlying connections are not noticed. Because a connection is not established, you must create a mirrored policy with a pass action for packets in the reverse direction.

You can also configure inspect parameters such as TCP thresholds and timeouts on a per-flow basis.

Zones and ACLs

Access control lists (ACLs) applied to interfaces that are members of zones are processed before the firewall policy is applied on the zone pair. You must ensure that interface ACLs do not interfere with the policy firewall traffic when there are policies between the source and destination zones. If a class map contains only an access list and does not contain a match protocol, a firewall attempts to match the flow protocol to known application-level gateways (ALGs) and process it as required.

Pinholes or ports opened through a firewall that allows applications-controlled access to a protected network are not punched for return traffic in interface ACLs.

Class Maps and Policy Maps for Zone-Based Policy Firewalls

Quality of service (QoS) class maps have numerous match criteria; firewalls have fewer match criteria. Firewall class maps are of type inspect and this information controls what shows up under firewall class maps.

A policy is an association of traffic classes and actions. It specifies what actions should be performed on defined traffic classes. An action is a specific function, and it is typically associated with a traffic class. For example, inspect, pass, and drop are actions.

Layer 3 and Layer 4 Class Maps and Policy Maps

Layer 3 and Layer 4 class maps identify traffic streams on which different actions should be performed.

A Layer 3 or Layer 4 policy map is sufficient for the basic inspection of traffic.

The following example shows how to configure class map c1 with the match criteria of ACL 101 and HTTP protocol. This command also creates an inspect policy map named p1 which specifies that the packets will be dropped as a part of the traffic at c1:

```
Device(config)# class-map type inspect match-all c1
Device(config-cmap)# match access-group 101
Device(config-cmap)# match protocol http
Device(config-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# drop
```

Note On Cisco ASR 1000 Series Aggregation Services Routers, the firewall supports a maximum of 1000 policy maps and 8 classes inside a policy map. You can configure a maximum of 16 match statements in a class map and 1000 globally.

Class-Map Configuration Restriction

If traffic meets multiple match criteria, these match criteria must be applied in the order of specific to less specific. For example, consider the following class map:

```
class-map type inspect match-any my-test-cmap
 match protocol http
 match protocol tcp
```

In this example, the **match protocol http** command is first applied to the HTTP traffic to ensure that the traffic is handled by the service-specific capabilities of HTTP inspection. If the match lines are reversed, and the the **match protocol tcp** command is applied to the traffic before the **match protocol http** command, the traffic is classified as TCP traffic and inspected according to the capabilities of the TCP inspection component of the firewall. If the match protocol TCP is configured first, it creates issues for services such as FTP and TFTP, and for multimedia and voice signaling services such as H.323, Real Time Streaming Protocol (RTSP), Session Initiation Protocol (SIP), and Skinny Client Control Protocol (SCCP). These services require additional inspection capabilities to recognize more complex activities.

Note Configure zone-based firewall on the device such that the TCP traffic flow does not exceed 65k in the window size.

Class-Default Class Map

In addition to user-defined classes, a system-defined class map named class-default represents all the packets that do not match any of the user-defined classes in a policy. The class-default class is always the last class in a policy map.

You can define explicit actions for a group of packets that does not match any of the user-defined classes. If you do not configure any actions for the class-default class in an inspect policy, the default action is drop.

Note For a class-default in an inspect policy, you can configure only drop action or pass action.

The following example shows how to use class-default class in a policy map. In this example, the HTTP traffic is dropped, and the remaining traffic is inspected. Class map c1 is defined for HTTP traffic, and class-default class is used for a policy map p1.

```
Device(config)# class-map type inspect match-all c1
Device(config-cmap)# match protocol http
Device(config-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class type inspect c1
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
```

```
Device(config-pmap-c)# drop
```

Supported Protocols for Layer 3 and Layer 4

The following protocols are supported:

- FTP
- H.323
- Real Time Streaming Protocol (RTSP)
- Skinny Client Control Protocol (SCCP)
- Session Initiation Protocol (SIP)
- Trivial File Transfer Protocol (TFTP)
- Route Convergence Monitoring and Diagnostics (RCMD)
- Lightweight Directory Access Protocol (LDAP)
- HTTP
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP/ESMTP)
- Post Office Protocol 3 (POP3)
- Internet Mail Access Protocol (IMAP)
- SUN Remote Procedure Call (SUNRPC)
- GPRS Tunnel Protocol version 0/1 (GTPv1)
- GPRS Tunnel Protocol version 2 (GTPv2)
- Point-to-Point Tunneling Protocol (PPTP)

Access Control Lists and Class Maps

Access lists are packet-classifying mechanisms. Access lists define the actual network traffic that is permitted or denied when an ACL is applied to a specific class map. Thus, the ACL is a sequential collection of permit and deny conditions that apply to a packet. A router tests packets against the conditions set in the ACL one at a time. A deny condition is interpreted as *do not match* Packets that match a deny access control entry (ACE) cause an ACL process to be terminated and the next match statement within the class to be examined.

Note You can configure the range of variables in an ACL as match criteria for a class map. Because the firewall supports only the 5-tuple match criteria, only source address, source port, destination address, destination port and protocol match criteria are supported. Any other match criteria that is configured and accepted by the CLI, is not supported by the firewall

Class maps are used to match a range of variables in an ACL, based on the following criteria:

- If a class map does not match a permit or a deny condition, then the ACL fails.

- The match-all or match-any condition is applied to the match statements contained within the class map. ACLs are processed as normal, and the result is used when comparing against match-all or match-any.
- If a match-all attribute is specified, and any match condition, ACL, or protocol fails to match the packet, further evaluation of the current class is stopped, and the next class in the policy is examined.
- If any match in a match-any attribute succeeds, the class-map criteria are met and the action that is defined in the policy is performed.
- If an ACL matches the match-any attribute, the firewall attempts to ascertain the Layer 7 protocol based on the destination port.

If you specify the match-all attribute in a class map, the Layer 4 match criteria (ICMP, TCP, and UDP) are set, but the Layer 7 match criteria is not set. Hence, the Layer 4 inspection is performed and Layer 7 inspection is omitted.

Access lists come in different forms—standard and extended access lists. Standard access lists are defined to permit or deny an IP address or a range of IP addresses. Extended access lists define both the source and the destination IP address or an IP address range. Extended access lists can also be defined to permit or deny packets based on ICMP, TCP, and UDP protocol types and the destination port number of the packet.

The following example shows how a packet received from the IP address 10.2.3.4 is matched with the class test1. In this example, the access list 102 matches the deny condition and stops processing other entries in the access list. Because the class map is specified with a match-all attribute, the class-map test1 match fails. However, the class map is inspected if it matches one of the protocols listed in the test1 class map.

If the class map test1 had a match-any attribute instead of match-all, the ACL would have matched deny and failed, but the ACL would have matched the HTTP protocol and performed the inspection using pmap1.

```
access-list 102 deny ip 10.2.3.4 0.0.0.0 any
access-list 102 permit any any
class-map type inspect match-all test1
 match access-list 102
 match protocol http
!
class-map type inspect match-any test2
 match protocol sip
 match protocol ftp
 match protocol http
!
parameter-map type inspect pmap1
 tcp idle-time 15
!
parameter-map type inspect pmap2
 udp idle-time 3600
!
policy-map type inspect test
 class type inspect test1
  inspect pmap1
!
 class type inspect test2
  inspect pmap2
!
 class type inspect class-default
  drop log
```

Hierarchical Policy Maps

A policy can be nested within another policy. A policy that contains a nested policy is called a hierarchical policy.

To create an hierarchical policy, attach a policy directly to a class of traffic. An hierarchical policy contains a child policy and a parent policy. The child policy is a previously defined policy that is associated with the new policy using the **service-policy** command. The new policy that uses the pre-existing policy is the parent policy.

Note There can be a maximum of two levels in an hierarchical inspect service policy.

For example, define two access lists—marketing and engineering. Create a class map that does a match against any of the two access groups. Then, create another class map that includes the previous class map with a match-all condition and match the protocol HTTP.

Parameter Maps

A parameter map allows you to specify the parameters that control the behavior of actions and match the criteria specified under a policy map and a class map, respectively.

There are two types of parameter maps:

- Inspect parameter map: An inspect parameter map is optional. If you do not configure a parameter map, the software uses default parameters. Parameters associated with the inspect action apply to all maps. If parameters are specified at both the top and lower levels, parameters at the lower levels override those in the top levels.
- Protocol-specific parameter map: A parameter map that is required for an Instant Messenger (IM) application (Layer 7) policy map.

Firewall and Network Address Translation

Network Address Translation (NAT) enables private IP internetworks that use nonregistered IP addresses to connect to the internet. NAT operates on a device, usually connecting two networks, and translates private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded to another network. NAT can be configured to advertise only one address for the entire network to the outside world. A device configured with NAT has at least one interface that connects to the inside network and one to the outside network.

In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet leaves the domain, NAT translates the locally significant source address to a global unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If the software cannot allocate an address because it has run out of addresses, it drops the packet and sends an ICMP host unreachable packet.

With reference to NAT, the term *inside* refers to those networks that are owned by an organization and that must be translated. Inside this domain, hosts will have addresses in one address space. When NAT is configured and when the hosts are outside, hosts will appear to have addresses in another address space. The inside address space is referred to as the local address space and the outside address space is referred to as the global address space.

Consider a scenario where NAT translates both source and destination IP addresses. A packet is sent to a device from inside NAT with the source address 209.168.1.1 and the destination address 10.1.1.1. NAT

translates these addresses and sends the packet to the external network with the source address 209.165.200.225 and the destination address 209.165.200.224.

Similarly, when the response comes back from outside NAT, the source address will be 209.165.200.225 and the destination address will be 209.165.200.224. Therefore, inside NAT, the packets will have a source address of 10.1.1.1 and a destination address of 209.168.1.1.

In this scenario, if you want to create an Application Control Engine (ACE) to be used in a firewall policy, the pre-NAT IP addresses (also known as inside local and outside global addresses) 209.168.1.1 and 209.165.200.224 must be used. In general, we do not recommend mapping outside global addresses.

WAAS Support for the Cisco Firewall

Depending on your release, the Wide Area Application Services (WAAS) firewall software provides an integrated firewall that optimizes security-compliant WANs and application-acceleration solutions with the following benefits:

- Integrates WAAS networks transparently.
- Protects transparent WAN-accelerated traffic.
- Optimizes a WAN through full stateful-inspection capabilities.
- Simplifies Payment Card Industry (PCI) compliance.
- Supports the Network Management Equipment-Wide Area Application Engine (NME-WAE) modules or standalone WAAS device deployment.

WAAS has an automatic discovery mechanism that uses TCP options during the initial three-way handshake to identify WAE devices transparently. After automatic discovery, optimized traffic flows (paths) experience a change in the TCP sequence number to allow endpoints to distinguish between optimized and nonoptimized traffic flows.

Note Paths are synonymous with connections.

WAAS allows the Cisco firewall to automatically discover optimized traffic by enabling the sequence number to change without compromising the stateful Layer 4 inspection of TCP traffic flows that contain internal firewall TCP state variables. These variables are adjusted for the presence of WAE devices.

If the Cisco firewall notices that a traffic flow has successfully completed WAAS automatic discovery, it permits the initial sequence number shift for the traffic flow and maintains the Layer 4 state on the optimized traffic flow.

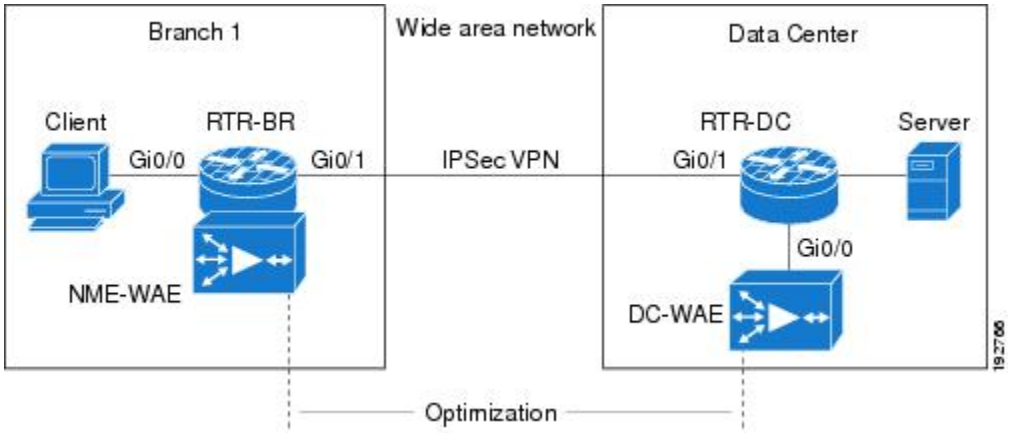
Note Stateful Layer 7 inspection on the client side can also be performed on nonoptimized traffic.

WAAS Traffic Flow Optimization Deployment Scenarios

The following sections describe two different WAAS traffic flow optimization scenarios for branch office deployments. WAAS traffic flow optimization works with the Cisco firewall feature on a Cisco Integrated Services Router (ISR). ZBF inspects the clear text after WAAS has unoptimized the packet.

The following figure shows an example of an end-to-end WAAS traffic flow optimization with the Cisco firewall. In this particular deployment, an NME-WAE is deployed on the same device as the Cisco firewall. Web Cache Communication Protocol (WCCP) is used to redirect traffic for interception.

Figure 53: End-to-End WAAS Optimization Path

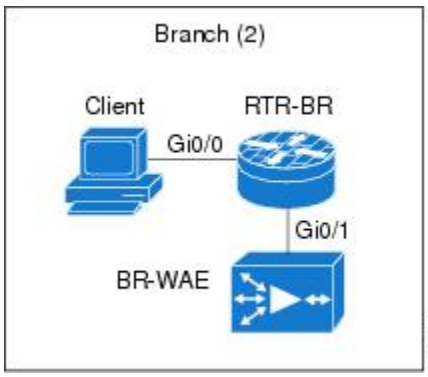


WAAS Branch Deployment with an Off-Path Device

A WAE device can be either a standalone WAE device or an NME-WAE device that is installed on an ISR as an integrated service engine as shown in the figure Wide Area Application Service [WAAS] Branch Deployment in this section.

The following figure shows a WAAS branch deployment that uses WCCP to redirect traffic to an off-path, standalone WAE device for traffic interception. The configuration for this option is the same as the WAAS branch deployment with an NME-WAE.

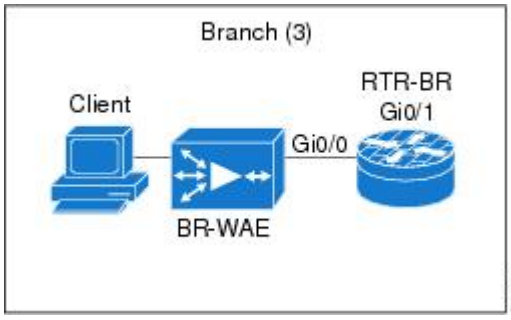
Figure 54: WAAS Off-Path Branch Deployment



WAAS Branch Deployment with an Inline Device

The following figure shows a WAAS branch deployment that has an inline WAE device that is physically in front of the Integrated Services Router (ISR). Because the WAE device is in front of the device, the Cisco firewall receives WAAS-optimized packets, and as a result, Layer 7 inspection on the client side is not supported.

Figure 55: WAAS Inline Path Branch Deployment



An edge WAAS device with the Cisco firewall is applied at branch office sites that must inspect the traffic moving to and from a WAN connection. The Cisco firewall monitors traffic for optimization indicators (TCP options and subsequent TCP sequence number changes) and allows optimized traffic to pass, while still applying Layer 4 stateful inspection and deep packet inspection to all traffic, and maintaining security while accommodating WAAS optimization advantages.

Note If the WAE device is in the inline location, the device enters the bypass mode after the automatic discovery process. Although the device is not directly involved in WAAS optimization, the device must be aware that WAAS optimization is applied to the traffic in order to apply Cisco firewall inspection to network traffic, and make allowances for optimization activity if optimization indicators are present.

Out-of-Order Packet Processing Support in the Zone-Based Firewalls

By default, the Cisco IOS XE firewall drops all out-of-order (OoO) packets when Layer 7 deep packet inspection is enabled or when Layer 4 inspection with Layer 7 protocol match is enabled. Dropping out-of-order packets can cause significant delays in end applications because packets are dropped only after the retransmission timer expires (on behalf of the sender). Layer 7 inspection is a stateful packet inspection and it does not work when TCP packets are out of order.

In Cisco IOS XE Release 3.5S, if a session does not require DPI, OoO packets are allowed to pass through the router and reach their destination. All Layer 4 traffic with OoO packets are allowed to pass through to their destination. However, if a session requires Layer 7 inspection, OoO packets are still dropped. By not dropping OoO packets when DPI is not required, the need to retransmit dropped packets and the bandwidth needed to retransmit on the network is reduced.

Severity Levels of Debug Messages

The severity level of debug messages specifies the types of issues for which a message is logged. While enabling firewall debugging, you can specify the level of messages that should be logged. The following table provides details about severity levels of debug messages.

Table 165: Severity Levels of Firewall Debug Messages

Trace Level	Severity Levels	Description
Critical	1	Applies to issues that make the zone-based policy firewall unusable or where the packets cannot be forwarded. This is the default. Examples of critical events are: • Back pressure triggered by the log mechanism. • Resource limit exceeded. • Memory allocation failure. • High-availability state not allowing new sessions.
Error	2	Applies to all error conditions and packet-drop conditions. Examples of error events are: • Synchronized (SYN) cookie: The number of maximum destination reached. • Not an initiator packet. • Could not send packets. • Application layer gateway (ALG) error condition.
Information	3	Applies to informational messages. Examples of information events are: • Packet drop because of incorrect policy configuration, zone-check failure, malformed packets, or hardcoded limit or threshold. • State machine transition. • Session or imprecise channel database information, search results, and so on. • Packet classification status or result. • Packet pass or drop status. • Session hit or miss. • Packet that is sent is a TCP reset (RST) packet. • SYN cookie event.
Detail	4	All log messages are printed. Examples of detailed events are: • Data structures. • Ternary content addressable memory (TCAM) search keys and result structure. • Firewall event details.

Smart Licensing Support for Zone-Based Policy Firewall

Zone-based policy firewall features for Cisco ASR 1000 Series Aggregation Services Routers are packaged separately from the security package, and hence, zone-based policy firewall requires a separate license to enable and disable features. Smart License support for zone-based firewall on ASR1000 feature implements support for smart licensing at a feature level for Cisco ASR 1000 Series Aggregation Services Routers through the Cisco UniversalK9 IOS software image.

The device need not be reloaded to enable the feature. Smart licensing is not turned on by default. Smart Licensing is toggled on or off globally through the **license smart enable** command or when configuring a zone-based policy firewall through the **zone security** command. The **show license all** command displays the status of smart license when smart licensing is implemented. The following is a sample output from the **show license all** command when smart licensing is enabled globally.

```
Device# show license all

License Store: Primary License Storage
StoreIndex: 0   Feature: internal_service                 Version: 1.0
        License Type: Evaluation
        License State: Active, In Use
            Evaluation total period: 1  day  0  hour
            Evaluation period left: 18 hours 57 minutes
            Period used: 5  hours 2  minutes
            Expiry date: Mar 18 2016 14:15:02
        License Count: Non-Counted
        License Priority: Low
License Store: Built-In License Storage
StoreIndex: 0   Feature: adventerprise                    Version: 1.0
        License Type: EvalRightToUse
        License State: Active, In Use
            Evaluation total period: 8  weeks 4  days
            Evaluation period left: 8  weeks 3  days
            Period used: 5  hours 13 minutes
            Transition date: May 16 2016 14:03:52
        License Count: Non-Counted
        License Priority: Low          <-- (CSL mode license)

Device(config)# license smart enable
Device(config)# zone security z1
Device(config)# exit
Device# show license all

Smart Licensing Status
----------------------
Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 65 days, 14 hours, 19 minutes, 47 seconds

License Usage
-------------

(ASR_1000_AdvEnterprise):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE
```

```
(ASR_1000_firewall):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE

Product Information
----------------------
UDI: PID:ASR1013,SN:NWG165000A9

Agent Version
---------------
Smart Agent for Licensing: 1.5.1_rel/29
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3
```

The following is a sample output when smart licensing is disabled:

```
Device(config)# no zone security z1
Device(config)# exit
Device# show license all

Smart Licensing Status
----------------------

Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 65 days, 14 hours, 18 minutes, 58 seconds

License Usage
--------------

(ASR_1000_AdvEnterprise):
  Description:
  Count: 1
  Version: 1.0
  Status: EVAL MODE

Product Information
--------------------
UDI: PID:ASR1013,SN:NWG165000A9

Agent Version
-------------
Smart Agent for Licensing: 1.5.1_rel/29
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3

Device(config)# no license smart enable
Device(config)# exit
Device# show license all

License Store: Primary License Storage
StoreIndex: 0   Feature: internal_service                 Version: 1.0
        License Type: Evaluation
        License State: Active, Not in Use, EULA accepted
            Evaluation total period: 1  day  0  hour
            Evaluation period left: 18 hours 54 minutes
            Period used: 5  hours 5  minutes
        License Count: Non-Counted
```

```
        License Priority: Low
License Store: Built-In License Storage
StoreIndex: 0   Feature: adventerprise                    Version: 1.0
        License Type: EvalRightToUse
        License State: Active, Not in Use, EULA accepted
            Evaluation total period: 8  weeks 4  days
            Evaluation period left: 8  weeks 3  days
            Period used: 5  hours 17 minutes
        License Count: Non-Counted
        License Priority: Low               <--- (back to CSL mode)
```

Zone-Based Firewall Reclassification

From Cisco IOS XE 17.6.1, you can configure ZBFW Session Reclassification. With the ZBFW Reclassification feature, policy configuration changes are applied on the existing firewall sessions. A given flow is reclassified when a packet is received from the session initiator on an established session.

The following are some examples where this can occur:

- Adding, deleting, or editing filters under a class map by:
  - Removing a match protocol.
  - Removing an access group.
  - Editing an Access Control Entry (ACE) under an access-group.
  - Editing an object group.
- Adding, deleting, or editing an Application Visibility and Control (AVC) policy.

Depending on the modifications to a policy, one of the following actions might occur:

- Inspect to drop: The existing session is torn down and the session is removed from the session table.
- Inspect to pass: The existing session is torn down because the zone-based firewall does not inspect the flow. However, in this scenario, the traffic continues to flow.
- Inspect to inspect: The existing session is moved under a new class map.
- Pass to inspect / Drop to inspect: The existing behavior continues, and the flow is blocked because mid-flow reclassification is not supported.

Note When there is a policy change, you cannot establish data during mid-flow.

Prerequisites for Zone-Based Policy Firewalls

Before you create zones, you should group interfaces that are similar when they are viewed from a security perspective.

Restrictions for Zone-Based Policy Firewalls

- In a Cisco Wide Area Application Services (WAAS) and Cisco IOS XE firewall configuration, all the packets processed by a WAE device must go over the Cisco IOS XE firewall in both directions to support the WCCP generic routing encapsulation (GRE) redirect. This situation occurs when a Layer 2 redirect is not available. If a Layer 2 redirect is configured on the WAE, the system defaults to the GRE redirect to continue to function.
- Zone-based firewall cannot interoperate with WAAS and WCCP, when WCCP is configured with Layer 2 redirect method.
- Zone-based firewall configuration cannot be applied on Bridge Domain Interfaces (BDI) that involves a Cisco Unity Express Virtual (vCUE) call flow.
- The self zone is the only exception to the default deny-all policy. All traffic to any router interface is allowed until traffic is explicitly denied.
- In a WAAS and Cisco IOS XE firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).
- WCCP traffic redirection does not work when the zone-based policy firewall that is enabled with generic GRE is configured on an ASR is configured with Cisco ISR-WAAS I/O modules. This configuration is a wide-area networking optimization solution. For WCCP traffic redirection to work, remove the zone-based policy firewall configuration from interfaces. If you are using a WAE device, WCCP traffic redirection works correctly.

  In the context of WAAS, generic GRE is an out-of-path deployment mechanism that helps to return packets from the WAAS WAE, through the GRE tunnel to the same device from which they were originally redirected, after completing optimization.
- Stateful inspection support for multicast traffic is not supported between any zones, including the self zone. Use Control Plane Policing for protection of the control plane against multicast traffic.
- When an in-to-out zone-based policy is configured to match the ICMP on a Windows system, the **traceroute** command works. However, the same configuration on an Apple system does not work because it uses a UDP-based traceroute. To overcome this issue, configure an out-to-in zone-based policy using the **icmp time-exceeded** and **icmp host unreachable** commands with the **pass** command (not the **inspect** command). This restriction applies to Cisco IOS XE Release 3.1S and earlier releases.
- ACLs are supported in a class map. However, the ACL-based packet count is disabled by default. Perfilter statistics is available in zone-based firewalls from Cisco IOS XE Release 3.13S and later releases.
- ACL statements using object groups are ignored for packets that are sent to a rendezvous point (RP) for processing.
- Bridge-domain interfaces do not support zone-based firewall inspection, including all Layer 4 and Layer 7 inspection.
- The ZBF cannot inspect traffic when NAT NVI is enabled on the device.
- When traffic enters a zone pair, the firewall examines the entire connection table and matches the traffic with any connection in the table even if the ingress interface does not match the zone pair. In this scenario, asymmetrically routed traffic on the firewall may drop packets, if the inspect action is configured.

In Cisco IOS XE Release 3.15S and later releases, zone-mismatch drop is configured in the class parameter map. If zone-mismatch drop is set, then the zones are checked against the original zones used when the packet is classified. If the zone is not part of the zone pair, the packet is dropped. If zone-mismatch drop is not set, then the zones are not checked.

- When ZBF is configured, all the interfaces that are a part of a zone pair much have RII configured. Interfaces that match the peer device must have the same RII configured. Additionally, flows that are initiated between two interfaces, where even one of the interface does not have an RII assigned, do not sync to the standby
- The zone-based firewall is supported with dynamic interfaces only in the default zone. These interfaces are created or deleted dynamically when traffic is tunneled IPsec or VPN secure tunnels. Virtual templates are used to support certain types of dynamic interfaces. For more information, see Virtual Interfaces as Members of Security Zones, on page 1423.
- To disable the zone-based firewall configurations that have been applied on the interfaces, use the **platform inspect disable-all** command. Similarly, to enable zone-based firewall on the interfaces, use the **no platform inspect disable-all** command.

  To verify if the **platform inspect disable-all** command has been applied, use the following **show running** configuration:

```
show run | sec disable
platform inspect disable-all
```

Note By default, zone-based firewall is always enabled.

- When the **droplog** command is configured under a user-defined class or the default class of a policy, disabling the logging of dropped packets by configuring the **drop** command does not stop the log messages. This is a known issue and the workaround is to configure the **nodroplog** command before configuring the **drop** command to stop the logging of messages. This issue applies to the **pass** command as well. The following example shows the issue:

```
! Logging of dropped packets is enabled by configuring the drop log command.
policy-map type inspect INT-EXT
 class type inspect INT-EXT
  pass
 class class-default
  drop log
!
```

  The following example shows the workaround:

```
! In this example, the no drop log command is configured before the drop command.
policy-map type inspect INT-EXT
 class type inspect INT-EXT
  pass
 class class-default
  drop log
  no drop log
  drop
!
```

- With the ZBFW Session Reclassification feature, mid-flow inspection is not supported for stateful traffic. For example, because of policy configuration changes, the action of an existing flow could change from drop to inspect. In this case, ZBFW does not inspect the existing flow.

- High availability is not supported for zone-based firewall policy reclassification.

How to Configure Zone-Based Policy Firewalls

The following sections provide information about the various tasks that comprise the zone-based policy firewalls configuration.

Configuring Layer 3 and Layer 4 Firewall Policies

Layer 3 and Layer 4 policies are *top-level* policies that are attached to the target (zone pair). Perform the following tasks to configure Layer 3 and Layer 4 firewall policies.

Configuring a Class Map for a Layer 3 and Layer 4 Firewall Policy

Use the following task to configure a class map for classifying network traffic.

Note You must perform at least one match step from step 4, 5, or 6.

When packets are matched to an access group, a protocol, or a class map, a traffic rate is generated for these packets. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match the filters in the configured policy, but match the session directly. The statistics related to subsequent packets are shown as part of the inspect action.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** [**match-any** | **match-all**] *class-map-name*
4. **match access-group** {*access-group* | **name** *access-group-name*}
5. **match protocol** *protocol-name* [**signature**]
6. **match class-map** *class-map-name*
7. **end**
8. **show policy-map type inspect zone-pair session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Device(config)# class-map type inspect match-all c1`	Creates a Layer 3 or Layer 4 inspect type class map and enters class-map configuration mode.
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Device(config-cmap)# match access-group 101`	Configures the match criterion for a class map based on the ACL name or number.
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol http`	Configures the match criterion for a class map on the basis of a specified protocol. • Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 6	**match class-map** *class-map-name* **Example:** `Device(config-cmap)# match class-map c1`	Specifies a previously defined class as the match criteria for a class map.
Step 7	**end** **Example:** `Device(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.
Step 8	**show policy-map type inspect zone-pair session** **Example:** `Device(config-cmap)# show policy-map type inspect zone-pair session`	(Optional) Displays Cisco stateful packet inspection sessions created because a policy map is applied on the specified zone pair. **Note** The information displayed under the **Class-map** field is the traffic rate (bits per second) of the traffic that belongs to the connection-initiating traffic only. Unless the connection setup rate is significantly high and is sustained for multiple intervals over which the rate is computed, no significant data is shown for the connection.

Creating a Policy Map for a Layer 3 and Layer 4 Firewall Policy

Use this procedure to create a policy map for a Layer 3 and Layer 4 firewall policy that will be attached to zone pairs.

If you are creating an inspect type policy map, note that only the following actions are allowed: drop, inspect, pass, and service-policy.

Note You must perform at least one step from step 5, 8, 9, or 10.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-name*
5. **inspect** [*parameter-map-name*]
6. **drop** [**log**]
7. **pass**
8. **service-policy type inspect** *policy-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Device(config-pmap)# class type inspect c1`	Specifies the traffic class on which an action is to be performed and enters the policy-map class configuration mode.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect inspect-params`	Enables Cisco stateful packet inspection.
Step 6	**drop** [**log**] **Example:** `Device(config-pmap-c)# drop`	(Optional) Drops packets that are matched with the defined class. **Note** The actions **drop** and **pass** are exclusive, and the actions **inspect** and **drop** are mutually exclusive. That is, you cannot specify both of them at the same time. Only one can be specified.

	Command or Action	Purpose
Step 7	**pass** **Example:** `Device(config-pmap-c)# pass`	(Optional) Allows packets that are matched with the defined class.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-pmap-c)# service-policy type inspect p1`	Attaches a firewall policy map to a zone pair.
Step 9	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Creating an Inspect Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **log** {**dropped-packets** {**disable** | **enable**} | **summary** [**flows** *number*] [**time-interval** *seconds*]}
5. **alert** {**on** | **off**}
6. **audit-trail** {**on** | **off**}
7. **dns-timeout** *seconds*
8. **icmp idle-timeout** *seconds*
9. **max-incomplete** {**low** | **high**} *number-of-connections*
10. **one-minute** {**low** | **high**} *number-of-connections*
11. **sessions maximum** *sessions*
12. **tcp finwait-time** *seconds*
13. **tcp idle-time** *seconds*
14. **tcp max-incomplete host** *threshold* [**block-time** *minutes*]
15. **tcp synwait-time** *seconds*
16. **tcp window-scale-enforcement loose**
17. **udp idle-time** *seconds*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**parameter-map type inspect** {*parameter-map-name* | **global** | **default**} **Example:** `Device(config)# parameter-map type inspect eng-network-profile`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters that pertain to the **inspect** action, and enters parameter map type inspect configuration mode.
Step 4	**log** {**dropped-packets** {**disable** | **enable**} | **summary** [**flows** *number*] [**time-interval** *seconds*]} **Example:** `Device(config-profile)# log summary flows 15 time-interval 30`	(Optional) Configures packet logging during the firewall activity. **Note** This command is visible in parameter map type inspect configuration mode only.
Step 5	**alert** {**on** | **off**} **Example:** `Device(config-profile)# alert on`	(Optional) Enables Cisco stateful packet inspection alert messages that are displayed on the console.
Step 6	**audit-trail** {**on** | **off**} **Example:** `Device(config-profile)# audit-trail on`	(Optional) Enables audit trail messages.
Step 7	**dns-timeout** *seconds* **Example:** `Device(config-profile)# dns-timeout 60`	(Optional) Specifies the domain name system (DNS) idle timeout (the length of time for which a DNS lookup session will be managed when there is no activity).
Step 8	**icmp idle-timeout** *seconds* **Example:** `Device(config-profile)# icmp idle-timeout 90`	(Optional) Configures the timeout for the ICMP sessions.
Step 9	**max-incomplete** {**low** | **high**} *number-of-connections* **Example:** `Device(config-profile)# max-incomplete low 800`	(Optional) Defines the number of existing half-open sessions that will cause the Cisco firewall to start and to stop deleting half-open sessions.
Step 10	**one-minute** {**low** | **high**} *number-of-connections* **Example:** `Device(config-profile)# one-minute low 300`	(Optional) Defines the number of new unestablished sessions that will cause the system to start deleting half-open sessions and stop deleting half-open sessions.
Step 11	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 200`	(Optional) Sets the maximum number of allowed sessions that can exist on a zone pair. Use this command to limit the bandwidth used by the sessions.

	Command or Action	Purpose
Step 12	**tcp finwait-time** *seconds* **Example:** `Device(config-profile)# tcp finwait-time 5`	(Optional) Specifies the length of time a TCP session will be managed for after the Cisco firewall detects a finish-exchange (FIN-exchange).
Step 13	**tcp idle-time** *seconds* **Example:** `Device(config-profile)# tcp idle-time 90`	(Optional) Configures the timeout for TCP sessions.
Step 14	**tcp max-incomplete host** *threshold* [**block-time** *minutes*] **Example:** `Device(config-profile)# tcp max-incomplete host 500 block-time 10`	(Optional) Specifies threshold and blocking time values for TCP host-specific Denial-of-Service (DoS) detection and prevention.
Step 15	**tcp synwait-time** *seconds* **Example:** `Device(config-profile)# tcp synwait-time 3`	(Optional) Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 16	**tcp window-scale-enforcement loose** **Example:** `Device(config-profile)# tcp window-scale-enforcement loose`	(Optional) Disables the window scale option check in the parameter map for a TCP packet that has an invalid window scale option under the zone-based policy firewall.
Step 17	**udp idle-time** *seconds* **Example:** `Device(config-profile)# udp idle-time 75`	(Optional) Configures an idle timeout threshold of UDP sessions that are going through the firewall.
Step 18	**end** **Example:** `Device(config-profile)# end`	Exits parameter map type inspect configuration mode and returns to privileged EXEC configuration mode.

Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

You need two security zones to create a zone pair. However, you can create only one security zone and use a system-defined security zone called *self*. Note that if you select a *self* zone, you cannot configure inspect policing.

A zone pair can have the same zone for source and destination zone. By default, traffic that stays within a zone is not inspected. In addition, there is the default zone (interfaces with no zone assignment) which can also be specified.

Use this process to complete the following tasks:

- Assign interfaces to security zones.
- Attach a policy map to a zone pair.

- Create at least one security zone.
- Define zone pairs.

Tip Before you create zones, think about what should constitute the zones. The general guideline is that you should group interfaces that are similar when they are viewed from a security perspective.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **description** *line-of-description*
5. **exit**
6. **interface** *type number*
7. **zone-member security** *zone-name*
8. **exit**
9. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self** | *default*] **destination** [**self** | *default* | *destination-zone-name*]
10. **description** *line-of-description*
11. **service-policy type inspect** *policy-map-name*
12. **platform inspect match-statistics per-filter**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 7	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all the traffic in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone a part of a zone pair to which you should apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self** | *default*] **destination** [**self** | *default* | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 10	**description** *line-of-description* **Example:** `Device(config-sec-zone-pair)# description accounting network to internet`	(Optional) Describes the zone pair.
Step 11	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, the traffic is dropped by default.
Step 12	**platform inspect match-statistics per-filter** **Example:**	Enables zone-based firewall per-filter statistics.

	Command or Action	Purpose
	`Device(config-sec-zone-pair)# platform inspect match-statistics per-filter`	**Note** To enable per-filter statistics on the device, do the following: • Reload the device. Or, • Remove all the service policies and reapply the changes to the statistics. To activate the **platform inspect match-statistics per-filter** command, reapply all service policies.
Step 13	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits the security zone-pair configuration mode and returns to the privileged EXEC mode.

Configuring NetFlow Event Logging

Global parameter maps are used for NetFlow event logging. With NetFlow event logging enabled, logs are sent to an off-box, high-speed log collector. By default, this functionality is not enabled. If this functionality is not enabled, the firewall logs are sent to a logger buffer located in the route processor or console.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-global**
4. **log dropped-packets**
5. **log flow-export v9 udp destination** *ipv4-address port*
6. **log flow-export template timeout-rate** *seconds*
7. **end**
8. **show parameter-map type inspect-global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables Privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-global** **Example:**	Configures a global parameter map and enters parameter-map type inspect configuration mode.

	Command or Action	Purpose
	`Device(config)# parameter-map type inspect-global`	
Step 4	**log dropped-packets** **Example:** `Device(config-profile)# log dropped-packets`	Enables logging for all the packets dropped by the firewall.
Step 5	**log flow-export v9 udp destination** *ipv4-address port* **Example:** `Device(config-profile)# log flow-export v9 udp destination 192.0.2.0 5000`	Enables NetFlow event logging and provides the collector's IP address and port.
Step 6	**log flow-export template timeout-rate** *seconds* **Example:** `Device(config-profile)# log flow-export template timeout-rate 5000`	Specifies the template timeout value.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show parameter-map type inspect-global** **Example:** `Device# show parameter-map type inspect-global`	Displays the global inspect-type parameter map information.

Configuring the Firewall with WAAS

Perform the following task to configure an end-to-end WAAS traffic flow optimization for the firewall that uses L2 to redirect traffic to a WAE device for traffic interception. When configuring WCCP in a ZBFW enviornment, either L2 or GRE encapsulation is used. However, in this scenario, L2 redirection is important because GRE is required for zone based firewall.

In Cisco IOS XE software, WAAS support is enabled by default and WAAS processing is discovered.

Note Configuring the firewall with WAAS (steps 5 to 13) is not required post Cisco IOS XE Release 3.5S. The commands in steps 5 to 12 have been deprecated after Cisco IOS XE Release 3.5S.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** *service-id*
4. **ip wccp** *service-id*
5. **log dropped-packets enable**
6. **max-incomplete low**
7. **max-incomplete high**

8. **class-map type inspect** *class-name*
9. **match protocol** *protocol-name* [**signature**]
10. **exit**
11. **policy-map type inspect** *policy-map-name*
12. **class class-default**
13. **class-map type inspect** *class-name*
14. **inspect**
15. **exit**
16. **exit**
17. **zone security** *zone-name*
18. **description** *line-of-description*
19. **exit**
20. **zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
21. **description** *line-of-description*
22. **exit**
23. **interface** *type number*
24. **description** *line-of-description*
25. **zone-member security** *zone-name*
26. **ip address** *ip-address*
27. **ip wccp** *service-id* {**group-listen** | **redirect** {**in** | **out**}}
28. **exit**
29. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*]
30. **service-policy type inspect** *policy-map-name*
31. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**ip wccp** *service-id* **Example:** `Device(config)# ip wccp 61`	Enters the WCCP dynamically defined service identifier number.
Step 4	**ip wccp** *service-id* **Example:** `Device(config)# ip wccp 62`	Enters the WCCP dynamically defined service identifier number.

	Command or Action	Purpose
Step 5	**log dropped-packets enable** **Example:** `Device(config-profile)# log dropped-packets enable`	
Step 6	**max-incomplete low** **Example:** `Device(config)# max-incomplete low 18000`	
Step 7	**max-incomplete high** **Example:** `Device(config)# max-incomplete high 20000`	
Step 8	**class-map type inspect** *class-name* **Example:** `Device(config)# class-map type inspect most-traffic`	Creates an inspect type class map for the traffic class and enters class-map configuration mode. **Note** The **class-map type inspect most-traffic** command is hidden.
Step 9	**match protocol** *protocol-name* [**signature**] **Example:** `Device(config-cmap)# match protocol http`	Configures the match criteria for a class map on the basis of a specified protocol. Only Cisco stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 10	**exit** **Example:** `Device(config-cmap)# exit`	Exits the class-map configuration mode and returns to the global configuration mode.
Step 11	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 12	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the matching of the system default class. • If the system default class is not specified, unclassified packets are matched.
Step 13	**class-map type inspect** *class-name* **Example:** `Device(config-pmap)# class-map type inspect most-traffic`	Specifies the firewall traffic (class) map on which an action is to be performed and enters policy-map class configuration mode.
Step 14	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables Cisco stateful packet inspection.
Step 15	**exit** **Example:**	Exits policy-map class configuration mode and returns to policy-map configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c)# exit`	
Step 16	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 17	**zone security** *zone-name* **Example:** `Device(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 18	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 19	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 20	**zone-pair security** *zone-pair name* [**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 21	**description** *line-of-description* **Example:** `Device(config-sec-zone)# description accounting network`	(Optional) Describes the zone pair.
Step 22	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 23	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 24	**description** *line-of-description* **Example:** `Device(config-if)# description zone interface`	(Optional) Describes an interface.
Step 25	**zone-member security** *zone-name* **Example:**	Assigns an interface to a specified security zone.

	Command or Action	Purpose
	`Device(config-if)# zone-member security zone1`	**Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except the traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone a part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 26	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 10.70.0.1 255.255.255.0`	Assigns an interface IP address for the security zone.
Step 27	**ip wccp** *service-id* {**group-listen** | **redirect** {**in** | **out**}} **Example:** `Device(config-if)# ip wccp 61 redirect in`	Specifies WCCP parameters on the interface.
Step 28	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 29	**zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**} **destination** [**self** | *destination-zone-name*] **Example:** `Device(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 30	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 31	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to privileged EXEC mode.

Configuring Zone-Based Firewall Reclassification

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | session-reclassify-allow}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**parameter-map type inspect** {*parameter-map-name* | **global** | session-reclassify-allow}	Enables a session reclassification by configuring the **session-reclassify-allow** attribute under the **parameter-map type inspect-global** mode. To disable this configuration, use the **no** form of the **session-reclassify-allow** command.

Configuration Examples for Zone-Based Policy Firewalls

The following sections provide examples relating to the configuration of zone-based policy firewalls.

Example: Configuring Layer 3 and Layer 4 Firewall Policies

The following example shows a Layer 3 or Layer 4 top-level policy. The traffic is matched to ACL 199 and deep-packet HTTP inspection is configured. Configuring the **match access-group 101** enables Layer 4 inspection. As a result, Layer 7 inspection is omitted unless the class map is of type match-all.

```
class-map type inspect match-all http-traffic
 match protocol http
 match access-group 101
!
policy-map type inspect mypolicy
 class type inspect http-traffic
  inspect
  service-policy http http-policy
```

Example: Creating an Inspect Parameter Map

The following sample configuration shows an inspect paramter map creation.

```
parameter-map type inspect eng-network-profile
 alert on
 audit-trail on
 dns-timeout 60
 icmp idle-timeout 90
 max-incomplete low 800
 one-minute low 300
 sessions maximum 200
 tcp finwait-time 5
```

```
 tcp idle-time 90
 tcp max-incomplete host 500 block-time 10
 tcp synwait-time 3
 udp idle-time 75
```

Example: Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

Example: Creating a Security Zone

The following example shows how to create security zone z1, which is called finance department networks, and security zone z2, which is called engineering services network:

```
zone security z1
 description finance department networks
!
zone security z2
 description engineering services network
```

Example: Creating Zone Pairs

The following example shows how to create zones z1 and z2 and specify that the firewall policy map is applied in zone z2 for traffic flowing between zones:

```
zone-pair security zp source z1 destination z2
service-policy type inspect p1
```

Example: Assigning an Interface to a Security Zone

The following example shows how to attach Ethernet interface 0 to zone z1 and Ethernet interface 1 to zone z2:

```
interface ethernet0
 zone-member security z1
!
interface ethernet1
 zone-member security z2
```

Example: Zone-Based Firewall Per-filter Statistics

The following configuration example shows how to prevent memory shortage when a large number of firewall filters are created. To prevent memory shortage, you can enable the zone-based firewall per-filter statistics with the **platform inspect match-statistics per-filter** command. In the example, for each filter (ACL or UDP), there are statistics available for the number of packets and the number of bytes traversed through zone-based firewall.

```
Device# show policy-map type inspect zone-pair ogacl_zp
 Zone-pair: ogacl_zp
  Service-policy inspect : ogacl_pm
Class-map: ogacl_cm (match-any)
  Match: access-group name ogacl
        xxx packets, xxx bytes
      Match: protocol udp
        xxx packets, xxx bytes
```

Note Per-filter statistics are available only for match-any filters and are not applicable for match-all cases.

Note For Cisco IOS XE 16.3 and Cisco IOS XE 16.4 releases, to enable per-filter statistics, either reload the device or remove the service-policies and then reapply the service policies on the zone pair before the **platform inspect match-statistics per-filter** command is activated.

For Cisco IOS XE 3.17 release, you must save the configuration and reload the system to activate this command.

Note Similarly, to disable per-filter statistics, either reload the device or remove the service-policies and then reapply the service policies on the zone pair.

To check the TCAM memory used in a device, use the **show platform hardware qfp active classification feature-manager shm-stats-counter** command.

```
Device# show platform hardware qfp active classification feature-manager shm-stats-counter
    Shared Memory Information:
    Total shared memory size: 16777216
    Used shared memory size: 14703656
```

Note If traffic drops or per-filter statistics counters are not displayed, then probabilty is the TCAM shared memory used is more than 75% of the total TCAM.

Note If the shared memory used in the device is more than 75% of the capacity, the following warning message is displayed :

```
%CPP_FM-3-CPP_FM_TCAM_WARNING: SIP1: cpp_sp_svr: TCAM limit exceeded: Already used 75 percent
shared memory for per-filter stats.
```

If the shared memory used in the device is 100%, the following warning message is displayed:

```
%CPP_FM-3-CPP_FM_TCAM_WARNING: SIP1: cpp_sp_svr: TCAM limit exceeded: Shared memory for
per-filter stats overflow!
```

Example: Configuring NetFlow Event Logging

The following example specifies how to configure netflow event logging.

```
parameter-map type inspect global
 log dropped-packets
 log flow-export v9 udp destination 192.0.2.0 5000
 log flow-export template timeout rate 5000
```

Example: Configuring the Cisco Firewall with WAAS

The following is an example of an end-to-end WAAS traffic flow optimization configuration for the firewall that uses WCCP to redirect traffic to a WAE device for traffic interception.

The following configuration example shows how to prevent traffic from being dropped between security zone members because the integrated-service-engine interface is configured on a different zone, and each security zone member is assigned an interface.

```
! Zone-based firewall configuration on your router.
ip wccp 61
ip wccp 62
parameter-map type inspect global
 log dropped-packets enable
 max-incomplete low 18000
 max-incomplete high 20000
!
class-map type inspect match-any most-traffic
 match protocol icmp
 match protocol ftp
 match protocol tcp
 match protocol udp
!
policy-map type inspect p1
 class type inspect most-traffic
  inspect
!
 class class-default
  drop
!
zone security in
!
zone security out
!
zone security waas
!
zone-pair security in-out source in destination out
 service-policy type inspect p1
!
zone-pair security out-in source out destination in
 service-policy type inspect p1
!
zone-pair security waas-out source waas destination out
 service-policy type inspect p1
!
zone-pair security in-waas source in destination waas
 service-policy type inspect p1
!
interface GigabitEthernet0/0
 description WAN Connection
 no ip dhcp client request tftp-server-address
 no ip dhcp client request router
 ip address dhcp
 ip wccp 62 redirect in
 ip wccp 61 redirect out
 ip flow ingress
 ip nat outside
 ip virtual-reassembly in
 ip virtual-reassembly out
 zone-member security out
 load-interval 30
 delay 30
 duplex auto
```

```
 speed auto
!
interface GigabitEthernet0/1
 description Clients
 ip address 172.25.50.1 255.255.255.0
 ip pim sparse-mode
 ip nat inside
 ip virtual-reassembly in
 zone-member security in
 ip igmp version 3
 delay 30
 duplex auto
 speed auto
!
interface Vlan1
 description WAAS Interface
 ip address 172.25.60.1 255.255.255.0
 ip wccp redirect exclude in
 ip nat inside
 ip virtual-reassembly in
 zone-member security waas
 load-interval 30
!
```

The following example shows the configuration on the WAE for zone-based firewall support. Note that this configuration cannot be done on the router, only on the WAE.

```
!Configuration on the WAE.
primary-interface Virtual 1/0
interface Virtual 1/0
 ip address 172.25.60.12 255.255.255.0
!
ip default-gateway 172.25.60.1
wccp router-list 1 172.25.60.1
wccp tcp-promiscuous service-pair 61 62
 router-list-num 1
 redirect-method gre
 egress-method ip-forwarding
 enable
!
```

Example: Configuring Firewall with FlexVPN and DVTI Under the Same Zone

The following example shows a firewall with FlexVPN and Dynamic Virtual Tunnel Interfaces (DVTI) configured under the same zone:

```
crypto ikev2 proposal PROP
 encryption 3des
 integrity sha256
 group 5
crypto ikev2 policy POL
 match fvrf any
 proposal PROP
crypto ikev2 keyring keyring1
 peer peer
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco
crypto ikev2 profile prof1
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback1
```

```
 keyring local keyring1
 no shutdown
 Virtual-Template 1
class-map type inspect match-any cmap
 match protocol icmp
 match protocol tcp
 match protocol udp
policy-map type inspect pmap
 class type inspect cmap
 inspect
 class class-default
 drop log
zone security in
zone security zone1
zone-pair security zp1 source zone1 destination in
 service-policy type inspect pmap
crypto ipsec profile ipsec1
 set ikev2-profile prof1
interface Loopback1
 ip address 51.1.1.1 255.255.255.0
interface Gi0/0/0.2
 encapsulation dot1q 2
 ip address 100.1.1.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.3
 encapsulation dot1q 3
 ip address 100.1.2.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.4
 encapsulation dot1q 4
 ip address 100.1.3.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.5
 encapsulation dot1q 5
 ip address 100.1.4.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.6
 encapsulation dot1q 6
 ip address 100.1.5.1 255.255.255.0
 zone-member security in
interface Virtual-Template1 type tunnel
 ip unnumbered loopback1
 zone-member security zone1
 tunnel source loopback1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec1
ip route 60.0.0.0 255.0.0.0 192.168.2.2
```

Example: Configuring Firewall with FlexVPN and DVTI Under Different Zones

The following example shows a firewall with FlexVPN and Dynamic Virtual Tunnel Interfaces (DVTI) configured under different zones.

```
crypto ikev2 proposal PROP
 encryption 3des
 integrity sha256
 group 5
crypto ikev2 policy POL
 match fvrf any
 proposal PROP
crypto ikev2 keyring keyring1
 peer peer1
 address 0.0.0.0 0.0.0.0
```

```
 pre-shared-key cisco1
crypto ikev2 keyring keyring2
 peer peer2
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco2
crypto ikev2 keyring keyring3
 peer peer3
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco3
crypto ikev2 keyring keyring4
 peer peer4
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco4
crypto ikev2 keyring keyring5
 peer peer5
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco5
crypto ikev2 profile prof1
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback1
 keyring local keyring1
 no shutdown
 Virtual-Template 1
crypto ikev2 profile prof2
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback2
 keyring local keyring2
 no shutdown
 Virtual-Template 2
crypto ikev2 profile prof3
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback3
 keyring local keyring3
crypto ikev2 profile prof4
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback4
 keyring local keyring4
 no shutdown
 Virtual-Template 4
crypto ikev2 profile prof5
 authentication remote pre-share
 authentication local pre-share
 match identity remote address 0.0.0.0
 match address local interface loopback5
 keyring local keyring5
 no shutdown
 Virtual-Template 5
class-map type inspect match-any cmap
 match protocol icmp
 match protocol tcp
 match protocol udp
policy-map type inspect pmap
 class type inspect cmap
 inspect
 class class-default
 drop log
```

```
zone security in
zone security zone1
zone security zone2
zone security zone3
zone security zone4
zone security zone5
zone-pair security zp1 source zone1 destination in
 service-policy type inspect pmap
zone-pair security zp2 source zone2 destination in
 service-policy type inspect pmap
zone-pair security zp3 source zone3 destination in
 service-policy type inspect pmap
zone-pair security zp4 source zone4 destination in
 service-policy type inspect pmap
zone-pair security zp5 source zone5 destination in
 service-policy type inspect pmap
crypto ipsec profile ipsec1
 set ikev2-profile prof1
crypto ipsec profile ipsec2
 set ikev2-profile prof2
crypto ipsec profile ipsec3
 set ikev2-profile prof3
crypto ipsec profile ipsec4
 set ikev2-profile prof4
crypto ipsec profile ipsec5
 set ikev2-profile prof5
interface Loopback1
 ip address 50.1.1.1 255.255.255.0
interface Loopback2
 ip address 50.1.2.1 255.255.255.0
interface Loopback3
 ip address 50.1.3.1 255.255.255.0
interface Loopback4
 ip address 50.1.4.1 255.255.255.0
interface Loopback5
 ip address 50.1.5.1 255.255.255.0
interface Gi0/0/0.2
 encapsulation dot1q 2
 ip address 100.1.1.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.3
 encapsulation dot1q 3
 ip address 100.1.2.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.4
 encapsulation dot1q 4
 ip address 100.1.3.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.5
 encapsulation dot1q 5
 ip address 100.1.4.1 255.255.255.0
 zone-member security in
interface Gi0/0/0.6
 encapsulation dot1q 6
 ip address 100.1.5.1 255.255.255.0
 zone-member security in
interface Virtual-Template1 type tunnel
 ip unnumbered loopback1
 zone-member security zone1
 tunnel source loopback1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec1
interface Virtual-Template2 type tunnel
 ip unnumbered loopback2
```

```
 zone-member security zone2
 tunnel source loopback2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec2
interface Virtual-Template3 type tunnel
 ip unnumbered loopback3
 zone-member security zone3
 tunnel source loopback3
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec3
interface Virtual-Template4 type tunnel
 ip unnumbered loopback4
 zone-member security zone4
 tunnel source loopback4
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec4
interface Virtual-Template5 type tunnel
 ip unnumbered loopback5
 zone-member security zone5
 tunnel source loopback5
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec5
ip route 60.0.0.0 255.0.0.0 192.168.2.2
```

Additional References for Zone-Based Policy Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support.

CHAPTER 123

Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall provides advanced traffic filtering or inspection of IPv4 packets. With IPv6 support, the zone-based policy firewall supports the inspection of IPv6 packets. Prior to IPv6 support, the firewall supported only the inspection of IPv4 packets. Only Layer 4 protocols, Internet Control Messaging Protocol (ICMP), TCP, and UDP packets are subject to IPv6 packet inspection.

This module describes the firewall features that are supported and how to configure a firewall for IPv6 packet inspection.

Restrictions for Zone-Based Policy Firewall IPv6 Support

The following functionalities are not supported:

- Application-level gateways (ALGs)
- Box-to-box high availability (HA)
- Distributed Denial-of-Service attacks
- Firewall resource management
- Layer 7 inspection
- Multicast packets
- Per-subscriber firewall or the broadband-based firewall
- Stateless Network Address Translation 64 (NAT64)
- VRF-Aware Software Infrastructure (VASI)
- Wide Area Application Services (WAAS) and Web Cache Communication Protocol (WCCP)

Information About IPv6 Zone-Based Firewall Support over VASI Interfaces

IPv6 Support for Firewall Features

The firewall features described in the table below are supported by IPv6 packet inspection:

Table 166: Firewall Features Supported on IPv6

Feature	Configuration Information
Class maps	*Zone-Based Policy Firewall* module.
Internet Control Message Protocol Version 6 (ICMPv6), TCP, and UDP protocols	• *Firewall Stateful Inspection of ICMP* module. • *Zone-Based Policy Firewall* module.
IP fragmentation	*Virtual Fragmentation Reassembly* module.
Intrachassis HA	—
Logging of error messages	*Zone-Based Policy Firewall* module.
Nested class maps	*Nested Class Map Support for Zone-Based Policy Firewall* module.
Out-of-order packet handling	The “Out-of-Order Packet Handling” section in the *Zone-Based Policy Firewall* module.
Parameter-maps—For inspect type parameter maps, the number of sessions defined in the parameter map will be cumulative for IPv4 and IPv6 sessions	*Zone-Based Policy Firewall* module.
Policy maps	*Zone-Based Policy Firewall* module.
Port-to-application mapping	—
Stateful Network Address Translation 64 (NAT64)	The *Stateful Network Address Translation 64* module in the *IP Addressing: NAT Configuration Guide*.
TCP SYN Cookie	*Configuring Firewall TCP SYN Cookie* module.
VPN routing and forwarding (VRF)-aware firewall	*VRF-Aware Cisco IOS XE Firewall* module.
Virtual fragmentation reassembly (VFR)	*Virtual Fragmentation Reassembly* module.
Zone, default zone, and zone pair	*Zone-Based Policy Firewall* module.

Dual-Stack Firewalls

A dual-stack firewall is a firewall running IPv4 and IPv6 traffic at the same time. A dual-stack firewall can be configured in the following scenarios:

- One firewall zone running IPv4 traffic and another running IPv6 traffic.
- IPv4 and IPv6 coexist when deployed with stateful Network Address Translation 64 (NAT64). In this scenario, the traffic flows from IPv6 to IPv4 and vice versa.
- The same zone pair allows both IPv4 and IPv6 traffic.

Firewall Actions for IPv6 Header Fields

The firewall actions for IPv6 header fields (in the order they are available in the IPv6 header) are described in the following table:

Table 167: IPv6 Header Fields

IPv6 Header Field	IPv6 Header Field Description	Firewall Action
Version	Similar to the Version field in the IPv4 packet header, except that this field lists number 6 for IPv6, instead of number 4 for IPv4.	Must be IPv6.
Traffic Class	Similar to the Type of Service (ToS) field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.	Not inspected.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.	Not inspected.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.	The firewall uses this field on a limited basis to calculate the length of some of the Layer 4 protocols, such as ICMP and TCP.
Next Header Length	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header Length field determines the type of information that follows the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or a UDP packet, or an extension header.	The firewall must recognize this field to create a session.

IPv6 Header Field	IPv6 Header Field Description	Firewall Action
Hop Limit	Similar to the Time-to-Live (TTL) field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the Hop Limit value by one. Because the IPv6 header does not have a checksum, the device can decrement the value without recalculating the checksum.	Not inspected.

IPv6 Firewall Sessions

To perform stateful inspection of traffic, the firewall creates internal sessions for each traffic flow. The session information includes IP source and destination addresses, UDP or TCP source and destination ports or ICMP types, the Layer 4 protocol type (ICMP, TCP, or UDP), and VPN routing and forwarding (VRF) IDs. For an IPv6 firewall, the source and the destination addresses contain 128 bits of the IPv6 address.

The firewall creates a TCP session after receiving the first packet when the packet matches the configured policy. The firewall tracks the TCP sequence numbers and drops the TCP packets whose sequence numbers are not within the configured range. Sessions are removed when the TCP idle timer expires or when a Reset (RST) or Finish-Acknowledge (FIN-ACK) packet is received with the appropriate sequence numbers.

The firewall creates UDP sessions when the first UDP packet that matches the configured policy arrives and removes sessions when the UDP idle timer expires. The firewall does not create TCP or UDP sessions for IPv6 packets with multicast IPv6 or unknown IPv6 addresses.

Firewall Inspection of Fragmented Packets

The firewall supports the inspection of fragmented IPv6 packets. IP fragmentation is the process of breaking up a single IP datagram into multiple packets of smaller size. In IPv6, end nodes perform a path maximum transmission unit (MTU) discovery to determine the maximum size of the packet that is to be sent and generate IPv6 packets with the fragment extension header for packets larger than the MTU size.

The firewall inspects fragmented packets by using Virtual Fragmentation Reassembly (VFR). VFR examines the fragment extension header for out-of-sequence fragments and puts them in the correct order for inspection. When you enable the firewall on an interface by adding the interface to a zone, VFR is configured automatically on the same interface. If you explicitly disable VFR, the firewall only inspects the first fragments with Layer 4 headers and passes the rest of the fragments without inspection.

The fragment extension header appears in the following order of headers:

- IPv6 header
- Hop-by-hop options header
- Destination options header
- Routing header
- Fragment extension header

Cisco Express Forwarding checks IPv6 packets with fragment extension headers so that the firewall need not do further checks before processing the packets.

ICMPv6 Messages

IPv6 uses ICMPv6 to perform diagnostic functions, error reporting, and neighbor discovery. ICMPv6 messages are grouped into informational and error messages.

The firewall inspects only the following ICMPv6 messages:

- ECHO REQUEST
- ECHO REPLY
- DESTINATION UNREACHABLE
- PACKET TOO BIG
- PARAMETER PROBLEM
- TIME EXCEEDED

Note Neighbor discovery packets are passed and not inspected by the firewall.

Firewall Support of Stateful NAT64

The zone-based policy firewall supports Stateful NAT64. Stateful NAT64 translates IPv6 packets into IPv4 packets and vice versa. When both the firewall and Stateful NAT64 are configured on a router, the firewall uses IP addresses in an access control list (ACL) to filter packets. However, ACL does not support a mix of IPv4 and IPv6 addresses. Before the firewall and Stateful NAT64 can work together, you must use an IPv6 ACL and the IPv4 address must be embedded in the IPv6 ACL.

Note You cannot use VRF along with a firewall and a Stateful NAT64 configuration because Stateful NAT64 is not VRF-aware.

When a firewall class map uses an ACL, the ACL must use the real IP addresses on the host to configure packet flows. If only a source or a destination address is needed, either the IPv4 address or the IPv6 address is used in the class map ACL. Before the packet flow can be filtered based on both the source and destination addresses, the IPv6 address must be used and the IPv4 address must be embedded in the ACL. The ACL has to use IPv6 addresses to filter Stateful NAT64 packets.

Note Stateless NAT64 with firewall is not supported.

Port-to-Application Mapping

Port-to-application mapping (PAM) allows you to customize TCP or UDP port numbers for network services or applications. The firewall uses PAM to correlate TCP or UDP port numbers to specific network services or applications. By mapping port numbers to network services or applications, an administrator can force firewall inspection on custom configurations that are not defined by using well known ports. Use the **ip port-map** command to configure PAM.

High Availability and ISSU

The IPv6 firewall supports Intrabox HA. Firewall sessions are synchronized to the standby Embedded Services Processors (ESP) for a switchover. In Service Software Upgrade (ISSU) is also supported by the IPv6 firewall.

Pass Action for a Traffic Class

In a firewall, a traffic class identifies a set of packets based on its contents. You can define a class and apply an action to the identified traffic that reflects a policy. An action is a specific functionality that is associated with a traffic class. You can configure inspect, drop, and pass actions for a class.

The pass action passes the traffic from one zone to another. When the pass action is configured, the firewall does not inspect the traffic; it passes the traffic. In the IPv6 firewall, you must explicitly configure the pass action for the return traffic by defining a zone pair and a policy map with pass action.

The following example shows how to configure the pass action for policy maps, outside-to-inside-policy, and inside-to-outside-policy for IPv6 traffic:

```
policy-map type inspect outside-to-inside-policy
 class type inspect ipv6-class
  pass (Defines pass action for the ipv6-class from the outside to the inside)
!
 class class-default
!
policy-map type inspect inside-to-outside-policy
 class type inspect ipv4-class
  inspect (Defines inspect action for ipv4-class)
 class type inspect v6_class
  pass (Defines pass action for ipv6-class from the inside to the outside)
 class class-default
!
!
zone security inside
!
zone security outside
!
zone-pair security in-out source inside destination outside
 service-policy type inspect inside-to-outside-policy
!
zone-pair security out-in source outside destination inside
 service-policy type inspect outside-to-inside-policy
```

How to Configure Zone-Based Policy Firewall IPv6 Support

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:** `Device(config)# ip port-map ftp port 8090 list ipv6-acl`	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:**	Sets permit conditions for an IPv6 access list.

	Command or Action	Purpose
	`Device(config-ipv6-acl)# permit ipv6 any any`	
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:** `Device(config-cmap)# match access-group name ipv6-acl`	Configures the match criteria for a class map on the basis of the specified ACL.
Step 17	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Applying Zones to Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ipv6 address** *ipv6-address/prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security z2`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:**	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-sec-zone)# exit`	
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*] **Example:** `Device(config)# zone-pair security in-2-out source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy`	Attaches a policy map to a top-level policy map.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 11	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface.
Step 12	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 13	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone member security z1`	Configures the interface as a zone member. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of the zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 15	**show policy-map type inspect zone-pair sessions** **Example:** `Device# show policy-map type inspect zone-pair sessions`	Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair. • The output of this command displays both IPv4 and IPv6 firewall sessions.

Example

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv4 address and vice versa:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:32847=>(209.165.201.2:21) ftp SIS_OPEN
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [37:84]

        Half-open Sessions
         Session 110D930C [2001:DB8:1::104]:32848=>(209.165.201.2:21) ftp SIS_OPENING
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv6 address:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:63=>[2001:DB8:2::102]:63 udp SIS_OPEN
          Created 00:00:02, Last heard 00:00:01
          Bytes sent (initiator:responder) [162:0]
```

Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

The following task configures an IPv6 firewall with Stateful NAT64 dynamic port address translation (PAT).

A PAT configuration maps multiple IPv6 hosts to a pool of available IPv4 addresses on a first-come first-served basis. The dynamic PAT configuration directly helps conserve the scarce IPv4 address space while providing connectivity to the IPv4 Internet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no ip address**
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **ipv6 address** *ipv6-address/prefix-length*
9. **ipv6 enable**
10. **nat64 enable**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **zone member security** *zone-name*
15. **negotiation auto**
16. **nat64 enable**
17. **exit**
18. **ipv6 access-list** *access-list-name*
19. **permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address*
20. **exit**
21. **ipv6 route** *ipv6-prefix/length interface-type interface-number*
22. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
23. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
24. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone member security z1`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::2/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 9	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 10	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.25 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security z2`	Attaches an interface to a security zone.

	Command or Action	Purpose
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 16	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 18	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-ipv4-pair`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 19	**permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address* **Example:** `Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:1::2 host 209.165:201.25`	Sets permit conditions for an IPv6 access list, a source IPv6 host address, and a destination IPv6 host address.
Step 20	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 21	**ipv6 route** *ipv6-prefix/length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0`	Establishes static IPv6 routes.
Step 22	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.
Step 23	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.25 209.165.201.125`	Defines a Stateful NAT64 IPv4 address pool.
Step 24	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload** **Example:**	Enables NAT64 PAT or overload address translation.

	Command or Action	Purpose
	`Device(config)# nat64 v6v4 list nat64-ipv6-any pool pool1 overload`	
Step 25	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Zone-Based Policy Firewall IPv6 Support

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring Zones and Applying Zones to Interfaces

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
```

```
Device(config-if)# zone member security z1
Device(config-if)# end
```

Example: Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

```
configure terminal
 ipv6 unicast-routing
 interface gigabitethernet 0/0/0
  no ip address
  zone member security z1
  negotiation auto
  ipv6 address 2001:DB8:1::2/96
  ipv6 enable
  nat64 enable
!
 interface gigabitethernet 0/0/1
  ip address 209.165.201.25 255.255.255.0
  zone member security z2
  negotiation auto
  nat64 enable
!
 ipv6 access-list ipv6-ipv4-pair
  permit ipv6 host 2001:DB8:1::2 host 209.165:201.25
!
 ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0
 ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841
 nat64 v4 pool pool1 209.165.201.25 209.165.201.125
 nat64 v6v4 list nat64-ipv6-any pool pool1 overload
```

Additional References for Zone-Based Policy Firewall IPv6 Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Stateful NAT64	Stateful Network Address Translation 64

Standards and RFCs

Standard/RFC	Title
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Zone-Based Policy Firewall IPv6 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 168: Feature Information for Zone-Based Policy Firewall IPv6 Support

Feature Name	Releases	Feature Information
Zone-Based Policy Firewall IPv6 Support	Cisco IOS XE Release 3.6S	The Zone-Based Policy firewall supports the inspection of IPv6 packets. The following commands were introduced or modified: **ip port-map** and **show policy-map type inspect zone-pair**.

CHAPTER 124

VRF-Aware Cisco IOS XE Firewall

The VRF-Aware Cisco IOS XE Firewall applies the Cisco IOS XE Firewall functionality to VPN Routing and Forwarding (VRF) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge routers. SPs provide managed services to small and medium business markets.

The VRF-Aware Cisco IOS XE Firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

The VRF-aware firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

Note Cisco IOS XE Releases do not support Context-Based Access Control (CBAC) firewalls.

Prerequisites for VRF-Aware Cisco IOS XE Firewall

- Understand Cisco IOS XE firewalls.
- Configure VRFs.

Restrictions for VRF-Aware Cisco IOS XE Firewall

- If two VPN networks have overlapping addresses, VRF-aware Network Address Translation (NAT) is required for them to support VRF-aware firewalls. NAT does not support inter-VRF routing. You can use the VRF-aware software infrastructure (VASI) for the inter-VRF routing functionality.

- You cannot apply per-VRF firewall policies if crypto tunnels that belong to multiple VPNs terminate on a single interface.
- Site-Site crypto maps on VASI interfaces are not supported on the following platforms:
  - Cisco 1000 Series Integrated Services Routers
  - Cisco 4000 Series Integrated Services Routers
  - Cisco 1000v Cloud Services Routers
- The same zone cannot be applied to interfaces that are configured on different VRFs.

Information About VRF-Aware Cisco IOS XE Firewall

VRF-Aware Cisco IOS XE Firewall

A VRF-aware firewall inspects IP packets that are sent or received within a VRF. VRF allows multiple instances of routing tables to coexist within a single router. This allows VPN segregation and the ability to have independent overlapping of IP address spaces. VRF allows traffic from the customers of one service provider to be isolated from another. The Cisco IOS XE VRF support splits the router into multiple routing domains, with each routing domain consisting of its own set of interfaces and routing and forwarding tables. Each routing domain is referenced by a unique identifier called the table ID. The global routing domain and the default routing domain (that is not associated with any VRF) is addressed with the table ID, zero. VRF supports overlapping of IP address space, thereby allowing the traffic from nonintersecting VRFs to have the same IP address.

The VRF-Aware Cisco IOS XE Firewall provides the following benefits:

- Scalable deployment—Scales to meet any network's bandwidth and performance requirements.
- VPN support—Provides a complete VPN solution based on Cisco IOS XE IPsec and other software-based technologies, including Layer 2 Tunneling Protocol (L2TP) tunneling, and quality of service (QoS).
- AIC support—Provides policy maps for the Internet Message Access Protocol (IMAP), Post Office Protocol 3 (POP3), Simple Mail Transfer Protocol (SMTP), and Sun Remote Procedure Call (SUN RPC)
- Allows users to configure a per-VRF firewall. The firewall inspects IP packets that are sent and received within a VRF. The firewall also inspects traffic between two different VRFs (intersecting VRFs).
- Allows SPs to deploy the firewall on the provider edge (PE) router.
- Supports overlapping IP address space, thereby allowing traffic from nonintersecting VRFs to have the same IP address.
- Supports VRF (not global) firewall command parameters and Denial-of-Service (DoS) parameters so that the VRF-aware firewall can run as multiple instances (with VRF instances) that are allocated to various VPN customers.
- Generates high-speed logging (HSL) messages that contain the VRF ID; however these messages are collected by a single collector.

The VRF-aware firewall allows you to limit the number of firewall sessions. If the firewall sessions are not limited, it would be difficult for VRFs to share router resources because one VRF may consume a maximum amount of resources, leaving few resources for other VRFs and thereby causing the denial of service to other VRFs.

Note On the Cisco ASR 1000 Series Aggregation Services Routers the firewall supports a maximum of 4000 VRFs.

Address Space Overlap

A VRF splits the device into multiple routing domains. Each of these routing domains contain their own set of interfaces and routing tables. A routing table is referenced by using a per-VRF unique table ID. Zero is the default global routing table ID that is not associated with a VPN routing and forwarding (VRF).

Nonintersecting VRFs are allowed to have overlapping address spaces (that is, the IP address of one VRF may be contained in others).

VRF

VPN routing and forwarding (VRF) allows multiple instances of routing tables to coexit within a single device. A VRF contains a template of a VRF table in a provider edge (PE) device.

The overlapping addresses, usually resulting from the use of private IP addresses in customer networks, are one of the major obstacles to the successful deployment of a peer-to-peer (P2P) VPN implementation. You can use the Multiprotocol Label Switching (MPLS) VPN technology to overcome the overlapping addresses issue.

Each VPN has its own routing and forwarding table in the device so that any customer or site that belongs to a VPN is provided access only to the set of routes contained within that table. Any PE device in the MPLS VPN network therefore contains a number of per-VPN routing tables and a global routing table that is used to reach other devices in the service provider (SP) network. Effectively, a number of virtual devices are created in a single physical device.

VRF-Lite

The VRF-Lite Aware Firewall feature, also called the VRF without MPLS-aware firewall, allows a firewall zone to be applied to non-MPLS-enabled VPN routing and forwarding (VRF) interfaces.

The VRF-Lite Aware Firewall feature enables a service provider (SP) to support two or more VPNs, in which IP addresses can be overlapped among VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be physical, such as Ethernet ports, or logical, such as VLAN switched virtual interfaces (SVIs). However, a Layer 3 interface cannot belong to more than one VRF at a time.

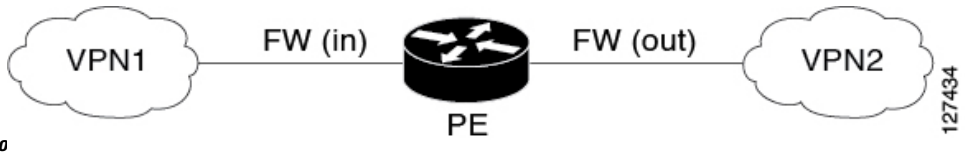
Note All VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes the following devices:

- Customer edge (CE) devices provide customers access to the SP network over a data link. The CE device advertises the site's local routes to the provider edge (PE) device and learns about the remote VPN routes from the PE device.
- PE devices exchange routing information with CE devices by using static routing or a routing protocol such as Border Gateway Protocol (BGP), Routing Information Protocol Version 1 (RIPv1), or RIPv2.
- PE devices (or core devices) are any devices in the SP network that are not attached to CE devices.
- A PE device is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE device to maintain all the SP VPN routes. Each PE device maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE device can be associated with a single VRF, if all of these sites are part of the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE devices, a PE device exchanges VPN routing information with other PE devices by using internal BGP (iBPG).

With VRF-lite, multiple customers can share one CE device, and only one physical link is used between the CE device and the PE device. The shared CE device maintains a separate VRF table for each customer, and switches or routes packets for each customer based on its own routing table. VRF-lite extends the limited PE device functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

Figure 56: Firewall in a VRF-to-VRF



Scenario

MPLS VPN

The Multiprotocol Label Switching (MPLS) VPN Feature allows multiple sites to interconnect transparently through a service provider (SP)network. One SP network can support several IP VPNs. Each VPN appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN.

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF consists of an IP routing table, a derived Cisco Express Forwarding table, and a set of interfaces that use the forwarding table.

The device maintains a separate routing and Cisco Express Forwarding table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems.

The device using Multiprotocol BGP (MP-BGP) distributes the VPN routing information using the MP-BGP extended communities.

VRF-Aware NAT

Network Address Translation (NAT) allows a single device, such as a device, to act as an agent between the Internet (or public network) and a local (or private) network. Although NAT systems can provide broad levels of security advantages, their main objective is to economize on address space.

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and need to access the Internet. Sites that do not possess Network Information Center (NIC)-registered IP addresses must acquire them. NAT eliminates the concern of NIC-registered IP addresses by dynamically mapping thousands of hidden internal addresses to a range of easy-to-get addresses.

A NAT system makes it difficult for an attacker to determine the following:

- Number of systems running on a network.
- Type of machines and operating systems running on the network.
- Network topology and arrangement.

NAT integration with Multiprotocol Label Switching (MPLS) VPNs allows multiple MPLS VPNs to be configured on a single device to work together. NAT can differentiate the MPLS VPNs from which it receives the IP traffic, even if all MPLS VPNs use the same IP addressing scheme. This enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

To provide value-added services, such as, Internet connectivity, domain name servers (DNS), and VoIP service to customers, MPLS service providers must use NAT. NAT helps MPLS VPN customers to use overlapped IP addresses in their network.

NAT can be implemented on a customer edge (CE) device or on a provider edge (PE) device. The NAT integration with MPLS VPNs feature enables the implementation of NAT on a PE device in an MPLS cloud.

VRF-Aware ALG

An application-layer gateway (ALG) is an application that translates the IP address information inside the payload of an application packet. The ALGs identify the address information in the packet payload that needs to be overwritten by NAT and supply the address information to NAT and firewall to create subordinate flows or doors to allow data to flow properly (an example of data flow is FTP data flow. Doors are transient structures that allow incoming traffic that matches a specific criterion. A door is created when there is not enough information to create a complete NAT session entry. A door contains information about the source and destination IP address and the destination port. However, it does not have information about the source port. When media data arrives, the source port information is known and the door is promoted to a real NAT session.

VRF-Aware IPsec

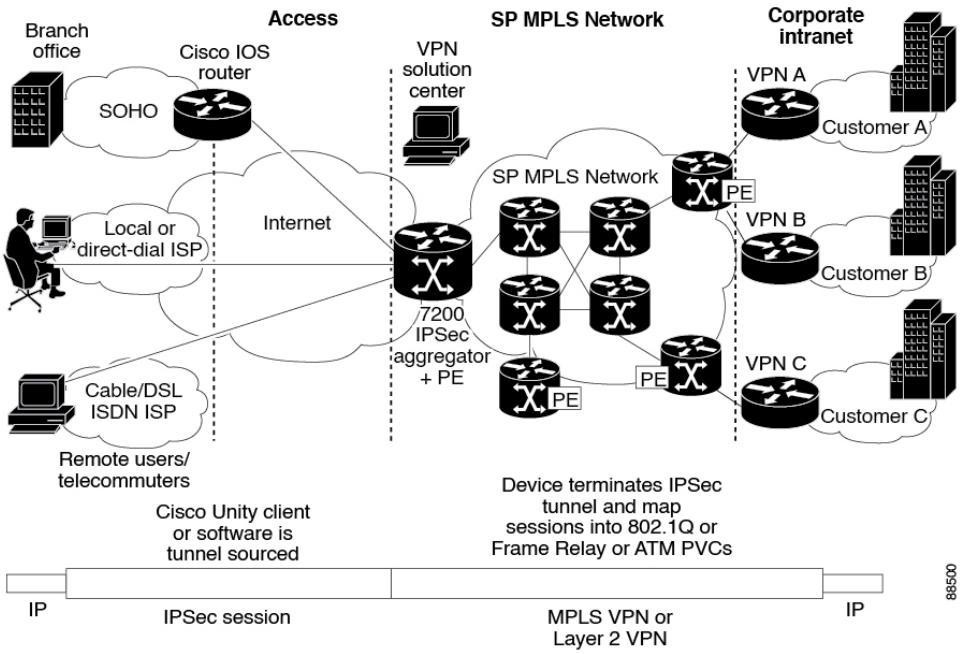
The VRF-Aware IPsec feature maps an IPsec tunnel to a Multiprotocol Label Switching (MPLS) VPN. Using the VRF-Aware IPsec feature, you can map IPsec tunnels to VPN routing and forwarding (VRF) instances using a single public-facing IP address.

Each IPsec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to a VRF domain called the Front Door VRF (FVRF). The inner, protected IP packet belongs to a domain called the Inside VRF (IVRF). In other words, the local endpoint of the IPsec tunnel belongs to the FVRF, whereas source and destination addresses of the inside packet belong to the IVRF.

One or more IPsec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

The following figure illustrates a scenario showing IPsec to MPLS and Layer 2 VPNs.

Figure 57: IPsec-to-MPLS and Layer 2 VPNs



VRF-Aware Software Infrastructure

The VRF-Aware Software Infrastructure (VASI) allows you to apply services such as access control lists (ACLs), NAT, policing, and zone-based firewalls to traffic that is flowing across two different VRF instances. The VASI interfaces support redundancy of the Route Processor (RP) and Forwarding Processor (FP). This feature supports IPv4 and IPv6 unicast traffic on VASI interfaces.

The primary use of VASI is to allow better isolation of VRFs. The VASI allows for per-VRF-specific features to be applied to the VASI interface without any impact to other VRFs that may share a common interface (for example, all VRFs may share the same interface to the Internet). For the firewall, this feature allows zones to be applied to the VASI.

VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF. The VASI virtual interface is the next hop interface for any packet that needs to be switched between these two VRFs. VASI interfaces provide the framework necessary to support NAT between two VRFs.

Each interface pair is associated with two different VRF instances. The two virtual interfaces, called vasileft and vasiright, in a pair are logically wired back-to-back and are completely symmetrical. Each interface has an index. The association of the pairing is done automatically based on the two interface indexes such that vasileft automatically gets paired to vasiright. You can configure either static routing or dynamic routing with BGP, Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF). BGP dynamic routing protocol restrictions and configuration are valid for BGP routing configurations between VASI interfaces. For more information on VASI, see the “ *Configuring the VRF-Aware Software Infrastructure* ” feature.

Security Zones

A security zone is a group of interfaces to which a policy can be applied.

Grouping interfaces into zones involves two procedures:

- Creating a zone so that interfaces can be attached to it.
- Configuring an interface to be a member of a given zone.

By default, traffic flows between interfaces that are members of the same zone.

When an interface is a member of a security zone, all traffic (except traffic going to the device or initiated by the device) between that interface and an interface in a different zone is dropped by default. To permit traffic to and from a zone-member interface and another interface, you must make that zone part of a zone pair, and apply a policy to that zone pair. If the policy permits traffic through inspect or pass actions, traffic can flow through the interface.

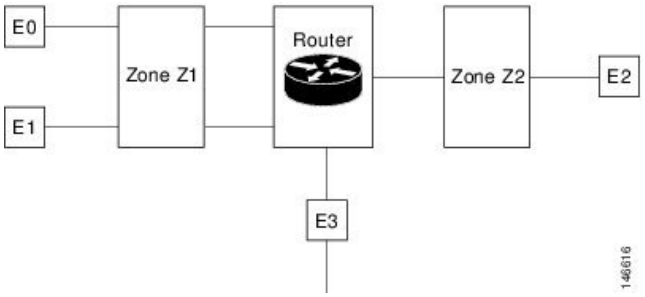
The following are the basic rules to consider when setting up zones:

- Traffic from a zone interface to a nonzone interface, or from a nonzone interface to a zone interface is always dropped; unless default zones are enabled (default zone is a nonzone interface).
- Traffic between two zone interfaces is inspected if there is a zone pair relationship for each zone, and if there is a configured policy for that zone pair.
- By default, all traffic between two interfaces in the same zone is always allowed.
- A zone pair can be configured with a zone as both source and destination zones. An inspect policy can be configured on this zone pair to inspect, pass, or drop the traffic between the two zones.
- An interface can be a member of only one security zone.
- When an interface is a member of a security zone, all traffic to and from that interface is blocked unless you configure an explicit interzone policy on a zone pair involving that zone.
- For traffic to flow between all the interfaces in a device, these interfaces must be members of one security zone or another. It is not necessary for all the device interfaces to be members of security zones.
- All the interfaces associated with a zone must be contained in the same virtual routing and forwarding (VRF).

Figure 1 illustrates the following:

- Interfaces E0 and E1 are members of security zone Z1.
- Interface E2 is a member of security zone Z2.
- Interface E3 is not a member of any security zone.

Figure 58: Security Zone Restrictions



- The zone pair and policy are configured in the same zone. Traffic flows freely between interfaces E0 and E1 because they are members of the same security zone (Z1).
- If no policies are configured, traffic will not flow between any other interfaces, for example, E0 and E2, E1 and E2, E3 and E1, and E3 and E2.
- Traffic can flow between E0 or E1 and E2 only when an explicit policy permitting traffic is configured between zone Z1 and zone Z2.
- Traffic can never flow between E3 and E0, E1, or E2 unless default zones are enabled.

Note On the Cisco ASR 1000 Series Aggregation Services Routers, the firewall supports a maximum of 4000 zones.

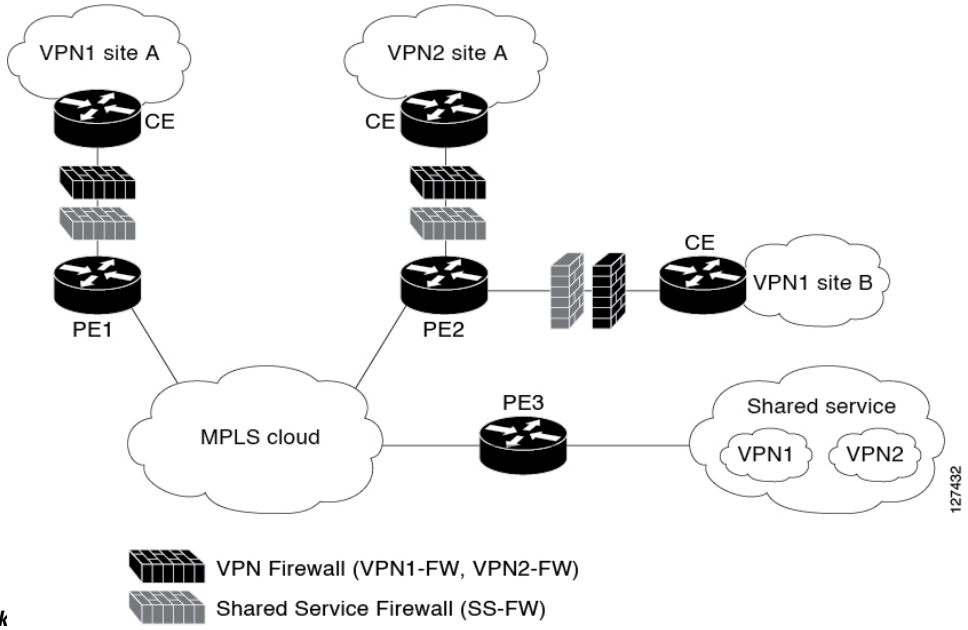
VRF-Aware Cisco Firewall Deployment

A firewall can be deployed at many points within the network to protect VPN sites from shared service (or the Internet) and vice versa. This section describes the following firewall deployment scenarios:

Distributed Network Inclusion of VRF-Aware Cisco Firewall

The following figure illustrates a typical situation in which a service provider (SP) offers firewall services to VPN customers VPN1 and VPN2, thereby protecting VPN sites from an external network (for example, shared services and the Internet) and vice versa.

Figure 59: Distributed



Network

In this example, VPN1 has two sites, Site A and Site B, that span across the Multiprotocol Label Switching (MPLS) core. Site A is connected to PE1, and Site B is connected to PE2. VPN2 has only one site that is connected to PE2. Each VPN has a VLAN segment in the shared service that is connected to the corresponding VLAN subinterface on PE3.

Each of the VPNs (VPN1 and VPN2) has two firewall rules—one to protect the VPN site from the shared service and another to protect the shared service from the VPN site. The firewall that protects the VPN site from the shared service is called the VPN firewall, and the firewall that protects the shared service from the VPN site is called the shared service firewall. Both firewall rules are applied on the VPN routing and forwarding (VRF) interface of each ingress provider edge (PE) device that is connected to the VPN site. The VPN firewall rule is applied in the ingress direction, because the VRF interface is ingress to the VPN site; and the shared service firewall rule is applied in the egress direction, because the VRF interface is egress to the shared service.

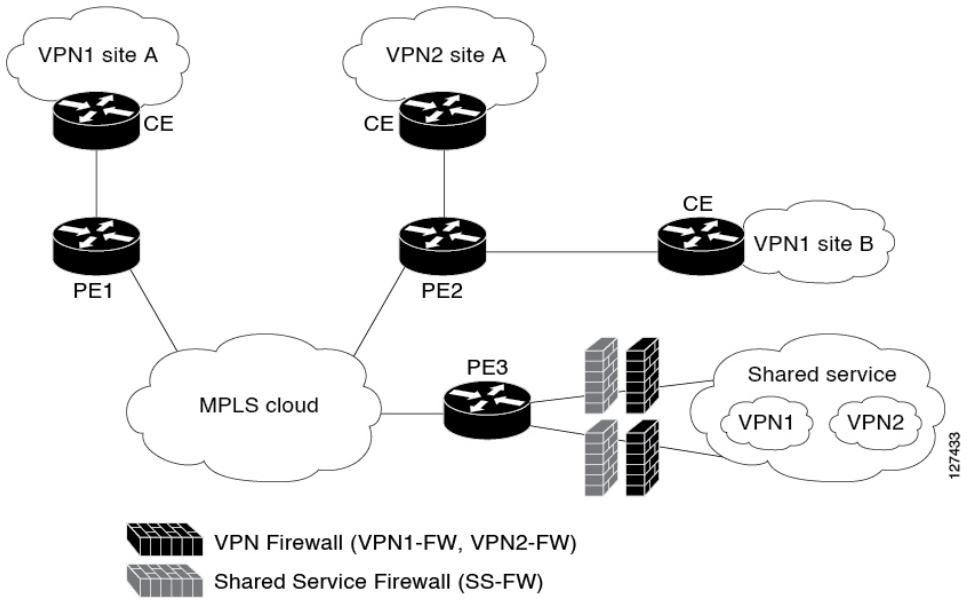
The benefits of using a distributed network are as follows:

- Because the firewall deployment is distributed across a Multiprotocol Label Switching (MPLS) cloud, the firewall processing load is distributed to all ingress PE devices.
- The shared service is protected from VPN sites at the ingress PE device, and hence malicious packets from VPN sites are filtered at the ingress PE device before they enter the MPLS cloud.
- VPN firewall features can be deployed in the ingress direction.

Hub-and-Spoke Network Inclusion of VRF-Aware Cisco Firewall

The following figure illustrates a hub-and-spoke network where firewalls for all VPN sites are applied on the egress PE device, PE3, which is connected to the shared service.

Figure 60: Hub-and-Spoke Network



Typically, each VPN has a VLAN and/or a VPN routing and forwarding (VRF) subinterface that is connected to the shared service. When a packet arrives at a Multiprotocol Label Switching (MPLS) interface, MPLS routes the packet to the corresponding subinterface that is connected to the shared service. Firewall policies on each VPN are applied on the corresponding subinterface (VRF interface) as shown in the above figure. The VPN firewall rule is applied in the egress direction because the subinterface is egress to the VPN site. And the shared service firewall rule is applied in the ingress direction because the subinterface is ingress to the shared service.

The benefits of a hub-and-spoke network are as follows:

- Because the firewall deployment is centralized to the egress provider edge (PE) device (PE3), deploying and managing the firewall is easy.
- The shared service firewall feature can be applied in the ingress direction.
- The VPN site is protected from the shared service at the egress PE device, and hence malicious packets from the shared service are filtered at the PE device before they enter the MPLS cloud.

How to Configure VRF-Aware Cisco IOS XE Firewall

Defining VRFs, Class Maps, and Policy Maps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*

5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **class-map type inspect match-any** *class-map-name*
9. **match protocol tcp**
10. **match protocol h323**
11. **exit**
12. **policy-map type inspect** *policy-map-name*
13. **class type inspect** *class-map-name*
14. **inspect** [*parameter-map-name*]
15. **exit**
16. **class class-default**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Defines a VRF instance and to enter VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 10:1`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 10:1`	Creates a route-target extended community for a VRF instance and exports routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 10:1`	Creates a route-target extended community for a VRF instance and imports routing information to the target VPN extended community.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 8	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any class-map1`	Creates a Layer 3 and Layer 4 (application-specific) inspect type class map and enters class-map configuration mode.
Step 9	**match protocol tcp** **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 10	**match protocol h323** **Example:** `Router(config-cmap)# match protocol h323`	Configures the match criterion for a class map on the basis of the specified protocol.
Step 11	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode and enters global configuration mode.
Step 12	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect global-vpn1-pmap`	Creates a Layer 3 and Layer 4 (protocol-specific) inspect type policy map and enters policy-map configuration mode.
Step 13	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect class-map1`	Specifies the traffic (class) on which an action is to be performed and enters policy-map-class configuration mode.
Step 14	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap-c)# inspect class-map1`	Enables Cisco IOS XE stateful packet inspection.
Step 15	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map-class configuration mode and enters policy-map configuration mode.
Step 16	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the default class so that you can configure or modify its policy. • The class-default class is defined by default. Configure the **class class-default** command to change the default drop attribute that is associated with the class-default.
Step 17	**end** **Example:** `Router(config-pmap)# end`	Exits policy-map configuration mode and enters global configuration mode.

Defining Zones and Zone Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone-name* **Example:** `Router(config)# zone security vpn1-zone`	Creates a security zone and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security global-zone`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:**	Creates a zone pair and enters security zone-pair configuration mode. • *zone-pair-name*--Name of the zone being attached to an interface.

	Command or Action	Purpose
	`Router(config)# zone-pair security vpn1-global-zone-pair source vpn1-zone destination global-zone`	• **source** *source-zone*--Specifies the name of the router from which traffic is originating. • **destination** *destination-zone*--Specifies the name of the router to which traffic is bound.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect global-vpn1-pmap`	Attaches a Layer 7 policy map to a top-level policy map.
Step 9	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits zone-pair configuration mode and enters privileged EXEC mode.

Applying Zones to Interfaces and Defining Routes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *name*
5. **ip address** *ip-address mask*
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **zone-member security** *zone-name*
12. **negotiation auto**
13. **exit**
14. **ip route vrf** *vrf-name destination-ip-address destination-prefix interface-type number* [**global**]
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip vrf forwarding** *name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security vpn1-zone`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/1/1`	Configures an interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.111.111.111 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security global-zone`	Attaches an interface to a security zone.
Step 12	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	**ip route vrf** *vrf-name destination-ip-address destination-prefix interface-type number* [**global**] **Example:** `Router(config)# ip route vrf vpn1 10.111.111.0 255.255.255.0 gigabitethernet 1/1/1 global`	Establishes static routes for a VRF instance.
Step 15	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for VRF-Aware Cisco IOS XE Firewall

Example: Defining VRFs, Class Maps, and Policy Maps

```
Router# configure terminal
Router(config)# ip vrf vrf1
Router(config-vrf)# rd 10:1
Router(config-vrf)# route-target export 10:1
Router(config-vrf)# route-target import 10:1
Router(config-vrf)# exit
Router(config)# class-map type inspect match-any class-map1
Router(config-cmap)# match protocol tcp
Router(config-cmap)# match protocol h323
Router(config-cmap)# exit
Router(config)# policy-map type inspect global-vpn1-pmap
Router(config-pmap)# class type inspect match-acl-111
Router(config-pmap-c)# inspect match-acl-111
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# end
```

Example: Defining Policy Maps, Zones, and Zone Pairs

```
Router# configure terminal
Router(config)# zone security vpn1-zone
Router(config-sec-zone)# exit
Router(config)# zone security global-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security vpn1-global-zone-pair source vpn1-zone destination
global-zone
Router(config-sec-zone-pair)# service-policy type inspect vpn1-global-pmap
Router(config-sec-zone-pair)# end
```

Example: Applying Zones to Interfaces and Defining Routes

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# zone-member security vpn1-zone
Router(config-if)# negotiation auto
Router(config-if)# exit
Router(config)# interface gigabitethernet 1/1/1
Router(config-if)# ip address 10.111.111.111 255.255.255.0
Router(config-if)# zone-member security global-zone
Router(config-if)# negotiation auto
Router(config-if)# exit
Router(config)# ip route vrf vpn1 10.111.111.0 255.255.255.0 gigabitethernet 1/1/1 global
Router(config)# end
```

Additional References for VRF-Aware Cisco IOS XE Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
NAT	Configuring Network Address Translation: Getting Started
MPLS VPN	*onfiguring a Basic MPLS VPN*
Zone-based Policy Firewall	*Zone-based Policy Firewall*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRF-Aware Cisco IOS XE Firewall

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 169: Feature Information for VRF-Aware Cisco IOS XE Firewall

Feature Name	Releases	Feature Information
VRF-Aware Cisco IOS XE Firewall	Cisco IOS XE Release 2.5	The VRF-Aware Cisco IOS XE Firewall feature applies the Cisco IOS XE Firewall functionality to VRF interfaces when the firewall is configured on an SP or large enterprise edge router.
Firewall--VRF-Aware ALG Support	Cisco IOS XE Release 2.5	The Firewall--VRF-Aware ALG Support feature allows ALG to extract the correct IP address and VRF ID from cached information when creating ALG tokens that require correct IP address VRF ID pairs.

Glossary

C3PL --Cisco Common Classification Policy Language. Structured, feature-specific configuration commands that use policy maps and class maps to create traffic policies based on events, conditions, and actions.

EHLO --Extended HELO substitute command for starting the capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by using the ESMTP protocol.

ESMTP --Extended Simple Mail Transfer Protocol. Extended version of the Simple Mail Transfer Protocol (SMTP), which includes additional functionality, such as delivery notification and session delivery. ESMTP is described in RFC 1869, SMTP Service Extensions.

HELO --Command that starts the SMTP capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by its fully qualified DNS hostname.

MAIL FROM --Start of an e-mail message that identifies the sender e-mail address (and name, if used), which appears in the From: field of the message.

MIME --Multipurpose Internet Mail Extension. Standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in e-mail, such as binary, foreign language text (such as Russian or Chinese), audio, or video data. MIME is defined in RFC 2045.

RCPT TO --Recipient e-mail address (and name, if used) that can be repeated multiple times for a likely message to deliver a single message to multiple recipients.

SMTP --Simple Mail Transfer Protocol. Internet protocol providing e-mail services.

CHAPTER 125

Layer 2 Transparent Firewalls

A Layer 2 transparent firewall operates on bridged packets and is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration.

This module provides an overview of the Layer 2 Transparent Firewalls feature.

Restrictions for Layer 2 Transparent Firewalls Support

- Address Resolution Protocol (ARP) inspection is not supported.
- Layer 2 forwarding technologies such as bridge domain, bridge domain interfaces (BDI), Overlay Transport Virtualization (OTV), X-Connect, Virtual Private LAN Services (VPLS), VxLAN, and non-IP flows, are not supported.
- Only normal IP or simple VLAN is supported on Ethernet frames. The transparent firewall generates TCP reset (RST) packets and sends these packets in supported Ethernet frame.
- TCP RST is not supported after intrabox high availability switchover.
- Virtual TCP (vTCP) is not supported.
- Network Address Translation (NAT), Box-to-Box (B2B) high availability, Multiprotocol Label Switching (MPLS), Virtual Routing and Forwarding (VRF) instances, VRF-Aware Software Infrastructure (VASI), Locator-ID Separation Protocol (LISP) are not supported in the Layer 2 switch path.
- Non IP packet flows like Ethernet Operation, Administration, and Maintenance (OAM), Connectivity Fault Management (CFM) is not supported.
- Layer 2-based access control lists (ACLs) are not supported in the transparent firewall class map.

Information About Layer 2 Transparent Firewalls

Layer 2 Transparent Firewall Support

A traditional zone-based firewall acts like a Layer 3 node in a network, and inspects the IP traffic that passes through the node. The traditional firewall is a routed hop and acts as a default gateway for hosts that connect to one of its screened subnets. However, to place this Layer 3 firewall in an existing network requires the network to be re-subnetted, which is time and resource-intensive. The Layer 2 transparent firewall is transparent to the network and does not require Layer 3 separation between segments. A transparent firewall acts like a "bump in the wire" or a "stealth firewall," and is not seen as a router hop to connected devices. Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network; IP readdressing is unnecessary. The transparent firewall operates on bridged packets and the Layer 3 firewall operates on routed packets.

A transparent firewall is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The transparent firewall only inspects IP packets.

A transparent firewall session is created by using IP Layer 3 and Layer 4 headers that contain 5-tuple information (5-tuple information are source and destination IP addresses, source and destination ports, and the protocol). The transparent firewall supports only Ethernet as a Layer 2 protocol, and supports both IPv4 and IPv6 addresses.

The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration. Both Layer 3 firewall and Layer 2 transparent firewall can coexist on a device.

The transparent firewall supports IP (Internet Control Message Protocol [ICMP], TCP, and UDP) inspection with the following topologies:

- Between two GigabitEthernet interfaces.
- Between a GigabitEthernet interface and a GigabitEthernet subinterface.
- Between two GigabitEthernet subinterfaces

The transparent firewall passes the following packets without a policy attached to them:

- Address Resolution Protocol (ARP)
- Multicast packets: Routing Information Protocol (RIP), Open Shortest Path First (OSPF), OSPF Version 3 (OSPFv3), Enhanced Interior Gateway Routing Protocol (EIGRP) IPv4 and IPv6 packets, Intermediate System-to-Intermediate System (ISIS) IPv4 and IPv6 packets
- Protocol-Independent Multicast (PIM) IPv4 and IPv6 packets
- Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and Gateway Load Balancing Protocol (GLBP)
- Internet Group Management Protocol (IGMP), and Multicast Listener Discovery (MLD)

How to Configure Layer 2 Transparent Firewalls

You can configure a Layer 2 transparent firewall using the same configuration as the zone-based firewalls. For more information, see the "Zone-Based Firewalls" module.

Configuration Examples for Layer 2 Transparent Firewalls

Example: Configuring a Layer 2 Transparent Firewall

The following example shows how to configure a Layer 2 transparent firewall with TCP and UDP inspection:

- Defines class maps.
- Defines policy maps.
- Defines zones and zone pairs.
- Attaches interfaces GigabitEthernet 0/0/0 and GigabitEthernet 0/0/1 to firewall zones.
- Enables local switching by connecting GigabitEthernet 0/0/0 with GigabitEthernet 0/0/1.

```
!Class map configuration
Device# configure terminal
Device(config)# class-map typ inspect match-any lan-wan-inspect-tcp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit
Device(config-cmap)# exit
Device(config)# class-map type inspect match-any wan-lan-inspect-udp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit

Device(config-cmap)# exit

!Policy map configuration
Device(config)# policy-map type inspect policy-wan-lan
Device(config-pmap)# class type inspect lan-wan-inspect-tcp
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# class type inspect wan-lan-inspect-udp
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# exit
Device(config-pmap)# exit

!Zones and zone pair configuration
Device(config)# zone security lan
Device(config-sec-zone)# exit
```

```
Device(config)# zone security wan
Device(config-sec-zone)# exit
Device(config)# zone-pair security lan2wan source lan destination wan
Device(config-sec-zone-pair)# service-policy type inspect policy-lan-wan
Device(config-sec-zone-pair)# exit
Device(config)# zone-pair security wan2lan source wan destination lan
Device(config-sec-zone-pair)# service-policy type inspect policy-wan-lan
Device(config-sec-zone-pair)# exit

! Interface configuration
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no ip address
Device(config-if)# zone-member security lan
Device(config-if)# exit
Device(config)# interface gigiabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# zone-member security wan
Device(config-if)# exit

!Local switching configuration
Device(config)# connect l2fw-conn gigabitethernet 0/0/0 gigabitethernet 0/0/1
Device(config)# end
```

Additional References for Layer 2 Transparent Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security Commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Zone-based firewalls	"Zone-Based Policy Firewalls" module in the *Zone-Based Policy Firewalls*, Configuration Guide.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Layer 2 Transparent Firewalls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 170: Feature Information for Layer 2 Transparent Firewalls

Feature Name	Releases	Feature Information
Layer 2 Transparent Firewalls	Cisco IOS XE 3.15S	A Layer 2 transparent firewall operates on bridged packets and is enabled on a pair of locally-switched Ethernet ports. Embedded IP packets forwarded through these ports are inspected similar to normal IP packets in a routing network. The zone-based firewall or Layer 3 firewall configuration can be applied to Layer 2 interfaces for the transparent firewall configuration. This feature is supported on Cisco ASR 1000 Series Aggregation Services Routers, and Cisco Cloud Services Router 1000V Series. No commands were introduced or updated for this feature.

CHAPTER 126

Nested Class Map Support for Zone-Based Policy Firewall

The Nested Class Map Support for Zone-Based Policy Firewall feature provides the Cisco IOS XE firewall the functionality to configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy. The Cisco IOS XE firewall supports up to three levels of class map hierarchy.

Prerequisites for Nested Class Map Support for Zone-Based Policy Firewall

Before configuring nested class maps, you should be familiar with the modular Quality of Service (QoS) CLI (MQC).

Information About Nested Class Map Support for Zone-Based Policy Firewall

Nested Class Maps

In Cisco IOS XE Release 3.5S and later releases, you can configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy. The nesting of class maps can be achieved by configuring the **match class-map** command. The only method

of combining the match-any and match-all characteristics within a single traffic class is by using the **class-map** command.

match-all and match-any Keywords of the class-map Command

To create a traffic class, you must configure the **class-map** command with the **match-all** and **match-any** keywords. You need to specify the **match-all** and **match-any** keywords only if more than one match criterion is configured in the traffic class. The following rules apply to the **match-all** and **match-any** keywords:

- Use the **match-all** keyword when all match criteria in the traffic class must be met to place a packet in the specified traffic class.
- Use the **match-any** keyword when only one of the match criterion in the traffic class must be met to place a packet in the specified traffic class.
- If you do not specify the **match-all** keyword or the **match-any** keyword, the traffic class behaves in a manner that is consistent with the **match-all** keyword.

Your zone-based policy firewall configuration supports nested class maps if the following criteria are met:

- Individual class maps in a hierarchy include multiple **match class-map** command references.
- Individual class maps in a hierarchy include match rules other than the **match class-map** command.

How to Configure Nested Class Map Support for Zone-Based Policy Firewall

Configuring a Two-Layer Nested Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **class-map match-any** *class-map-name*
7. **match protocol** *protocol-name*
8. **exit**
9. **class-map match-any** *class-map-name*
10. **match class-map** *class-map-name*
11. **match class-map** *class-map-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map match-any child1`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criteria for a class map on the basis of a specified protocol.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode and enters global configuration mode.
Step 6	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map match-any child2`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.
Step 7	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol udp`	Configures the match criteria for a class map on the basis of a specified protocol.
Step 8	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode and enters global configuration mode.
Step 9	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map match-any parent`	Creates a Layer 3 or Layer 4 class map and enters class map configuration mode.
Step 10	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map child1`	Configures a traffic class as a classification policy.
Step 11	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map child2`	Configures a traffic class as a classification policy.

	Command or Action	Purpose
Step 12	**end** **Example:** `Router(config-cmap)# end`	Exits class map configuration mode and enters privileged EXEC mode.

Configuring a Policy Map for a Nested Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class-type inspect** *class-map-name*
5. **inspect**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters policy map configuration mode.
Step 4	**class-type inspect** *class-map-name* **Example:** `Router(config-pmap)# class-type inspect parent`	Specifies the traffic (class) on which an action is to be performed and enters policy-map class configuration mode.
Step 5	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and enters privileged EXEC mode.

Attaching a Policy Map to a Zone Pair

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *zone-name* **destination** [*zone-name*]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Router(config)# zone security source-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Router(config)# zone security destination-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *zone-name* **destination** [*zone-name*]]	Creates a zone pair and enters security zone pair configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# zone-pair security secure-zone source source-zone destination destination-zone`	• To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect pmap`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security source-zone`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Nested Class Map Support for Zone-Based Policy Firewall

Example: Configuring a Two-Layer Nested Class Map

```
Router# configure terminal
Router(config)# class-map match-any child1
Router(config-cmap)# match protocol tcp
Router(config-cmap)# exit
Router(config)# class-map match-any child2
Router(config-cmap)# match protocol udp
Router(config-cmap)# exit
Router(config)# class-map match-any parent
Router(config-cmap)# match class-map child1
```

```
Router(config-cmap)# match class-map child2
Router(config-cmap)# end
```

Example: Configuring a Policy Map for a Nested Class Map

```
Router# configure terminal
Router(config)# policy-map type inspect pmap
Router(config-pmap)# class-type inspect parent
Router(config-pmap-c)# inspect
Router(config-pmap-c)# end
```

Example: Attaching a Policy Map to a Zone Pair

```
Router# configure terminal
Router(config)# zone security source-zone
Router(config-sec-zone)# exit
Router(config)# zone security destination-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security secure-zone source source-zone destination destination-zone
Router(config-sec-zone-pair)# service-policy type inspect pmap
Router(config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# zone-member security source-zone
Router(config-if)# end
```

Additional References for Nested Class Map Support for Zone-Based Policy Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Zone-based policy firewall	*Zone-Based Policy Firewall*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Nested Class Map Support for Zone-Based Policy Firewall

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 171: Feature Information for Nested Class Map Support for Zone-Based Policy Firewall

Feature Name	Releases	Feature Information
Nested Class Map Support for Zone-Based Policy Firewall	Cisco IOS XE Release 3.5S	The Nested Class Map Support for Zone-Based Policy Firewall feature provides the Cisco IOS XE firewall the functionality to configure multiple traffic classes (which are also called nested class maps or hierarchical class maps) as a single traffic class. When packets meet more than one match criterion, you can configure multiple class maps that can be associated with a single traffic policy.

CHAPTER 127

Zone Mismatch Handling

The Zone Mismatch Handling feature allows you to validate the zone pair that is associated with an existing session and allows traffic that matches the zone pair into the network. Allowing traffic into the network without validating the zone pair associated with a session can lead to security vulnerabilities.

This module provides an overview of the feature and explains how to configure it.

Restrictions for Zone Mismatch Handling

You cannot configure the **zone-mismatch drop** command under the **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, and **parameter-map type inspect global** commands.

Information About Zone Mismatch Handling

Zone Mismatch Handling Overview

The zone-based firewall creates sessions for traffic that flows from a source zone to a destination zone, and also matches the traffic when it returns from the destination zone to the source zone. A zone is a group of interfaces that have similar functions or features. A zone pair allows you to specify a unidirectional firewall policy between two security zones that are part of a zone pair.

For the first packet of the traffic, the firewall checks the zone pair that is associated with the ingress and egress interfaces of the packet, and validates the packet before it creates a session for traffic that can be inspected. And when the return traffic comes, the firewall does a session lookup based on the first packet to find an existing session. If the firewall finds a matching session, it allows the traffic to passthrough, and does not check whether the zone associated with the return traffic matches with the zone pair associated with the existing session. Allowing traffic into the network without validating the zone-pair associated with a session can lead to security vulnerabilities.

The Zone Mismatch Handling feature allows you to validate the zone pair that is associated with an existing session and allows traffic that matches the zone pair into the network. When you configure the **zone-mismatch drop** command, the firewall drops all packets (IPv4 and IPv6) that match an existing session but whose zone pair does not match the zone through which these packets arrive or leave. This feature works along with high availability and In-Service Software Upgrade (ISSU).

When you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global** command, the zone mismatch handling configuration applies to the global firewall configuration. Traffic between all zones are inspected for zone-pair mismatch.

You can also configure the **zone-mismatch drop** command under the **parameter-map type inspect** command. This allows you to apply the Zone-Mismatch Handling feature on a per-policy basis.

When you configure the **zone-mismatch drop** command, the configuration is effective only for new sessions. For existing sessions, traffic is not dropped if the sessions do not belong to the same zone-pair.

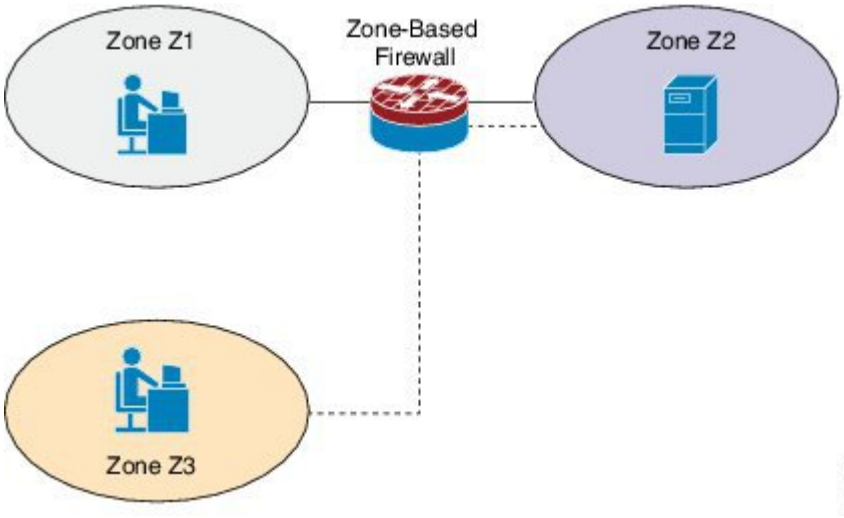
Deployment Scenarios for Zone Mismatch Handling

This section describes some typical scenarios in which the Zone Mismatch Handling feature is deployed:

Traffic Inspection by the Zone-Based Firewall

The following illustration shows traffic inspection by the firewall when the Zone Mismatch Handling feature is enabled.

Figure 61: Traffic Inspection by the Zone-Based Firewall



Zones Z1 and Z2 are part of the same zone pair, which has a parameter map that has the **zone-mismatch drop** command configured on it. Because zone Z3 is not part of the zone pair, the traffic from Z3 is dropped even if the traffic matches the firewall sessions between interface 1 and interface 2.

If you configure the **zone-mismatch drop** command for the parameter-map that is associated with the zone pair to which zone Z3 is attached, that configuration will not be effective for sessions established between Z1 and Z2. However, if you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global** command, the configuration is effective for traffic between all the zones.

Application Layer Gateways Configured with the Zone-Based Firewall

Some application layer gateways (ALGs) also called application-level gateways require multiple control and media channels to operate. The zone-based firewall does not enforce that control and media channels should be in the same zone pair for ALGs. When you configure the **zone-mismatch drop** command for media or data channels, the configuration takes effect after the media or data channels are promoted from imprecise to precise sessions. The zone-based firewall checks these precise sessions like normal sessions. Imprecise sessions are sessions that do not have all 5-tuple information.

How to Configure Zone Mismatch Handling

Configuring Zone Mismatch Handling

You cannot configure the zone-mismatch drop command under the **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, and **parameter-map type inspect global** commands.

If you configure the **zone-mismatch drop** command under the **parameter-map type inspect-global**command, the zone mismatch handling configuration applies to the global firewall configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **parameter-map type inspect** *parameter-map-name*
   - **parameter-map type inspect-global**
4. **zone-mismatch drop**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables user EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **parameter-map type inspect** *parameter-map-name* • **parameter-map type inspect-global** **Example:**	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the inspect action and enters parameter-map type inspect configuration mode.

	Command or Action	Purpose

	`Device(config)# parameter-map type inspect pmap1`
	or
	`Device(config)# parameter-map type inspect-global`	

Step 4	**zone-mismatch drop** **Example:** `Device(config-profile)# zone-mismatch drop`	Validates the zone pair that is attached to an existing session and allows traffic that matches the zone pair into the network. If the zone pair of an incoming session does not match the zone through which the session arrives or leaves, the firewall drops these packets.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Configuration Examples for Zone Mismatch Handling

Example: Configuring Zone Mismatch Handling

In the following example, the Zone Mismatch Handling feature is enabled for parameter map pmap-fw.

```
! Configuring zones
Device(config)# zone security private
Device(config-sec-zone)# exit
Device(config)# zone security public
Device(config-sec-zone)# exit
Device(config)# zone security internet
Device(config-sec-zone)# exit

! Attaching zones to interfaces
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# zone-member security private
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/6
Device(config-if)# ip address 209.165.200.226 255.255.255.0
Device(config-if)# zone-member security public
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/1
Device(config-if)# ip address 198.51.100.1 255.255.255.0
Device(config-if)# zone-member security internet
Device(config-if)# no shutdown
Device(config-if)# exit

!Configuring the Zone Mismatch Handling feature
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# zone-mismatch drop
Device(config-profile)# exit

!Configuring class maps
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol tcp
```

```
Device(config-cmap)# match protocol udp
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit

! Configuring policy maps and class matching
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# exit
Device(config-pmap)# exit

! Configuring zone pairs
Device(config)# zone-pair security private-internet source private destination internet
Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy
Device(config-sec-zone-pair)# end
```

Additional References for Zone Mismatch Handling

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security Commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Zone Mismatch Handling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 172: Feature Information for Zone Mismatch Handling

Feature Name	Releases	Feature Information
Zone Mismatch Handling	Cisco IOS XE 3.15S	The Zone Mismatch Handling feature allows you to validate the zone-pair associated with an existing session and allows traffic that matches the zone-pair into the network. This feature is supported on Cisco 4400 Series Integrated Services Routers,Cisco ASR 1000 Series Aggregation Services Routers, and Cisco Cloud Services Router 1000V Series. The following command was introduced: **zone-mismatch handling**.

CHAPTER 128

Configuring Firewall Stateful Interchassis Redundancy

The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of routers to act as backup for each other. This feature can be configured to determine the active router based on a number of failover conditions. When a failover occurs, the standby router seamlessly takes over and starts performing traffic forwarding services and maintaining a dynamic routing table.

Prerequisites for Firewall Stateful Interchassis Redundancy

- The interfaces attached to the firewall must have the same redundant interface identifier (RII).
- The active device and the standby device must have the same Cisco IOS XE Zone-Based Firewall configuration.
- The active device and the standby device must run on an identical version of the Cisco IOS XE software. The active device and the standby device must be connected through a switch.
- Embedded Service Processor (ESP) must match on both active and standby devices.

Restrictions for Firewall Stateful Interchassis Redundancy

- LAN and MESH scenarios are not supported.
- Cisco ASR 1006 and Cisco ASR 1013 platforms with dual Embedded Services Processors (ESPs) or dual Route Processors (RPs) in the chassis are not supported, because coexistence of interbox high availability (HA) and intrabox HA is not supported.

Cisco ASR 1006 and Cisco ASR 1013 platforms with single ESP and single RP in the chassis supports interchassis redundancy.

- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall Stateful Interchassis Redundancy configuration.

Information About Firewall Stateful Interchassis Redundancy

How Firewall Stateful Inter-Chassis Redundancy Works

You can configure pairs of routers to act as hot standbys for each other. This redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups. The figure below depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that has one outgoing interface. The *Redundancy Group Configuration--Two Outgoing Interfaces* figure depicts the active-active device scenario shows how two redundancy groups are configured for a pair of routers that have two outgoing interfaces.

Note that in both cases, the redundant routers are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and to synchronize the stateful database for these applications.

Also, in both cases, the pairs of redundant interfaces are configured with the same unique ID number known as the RII.

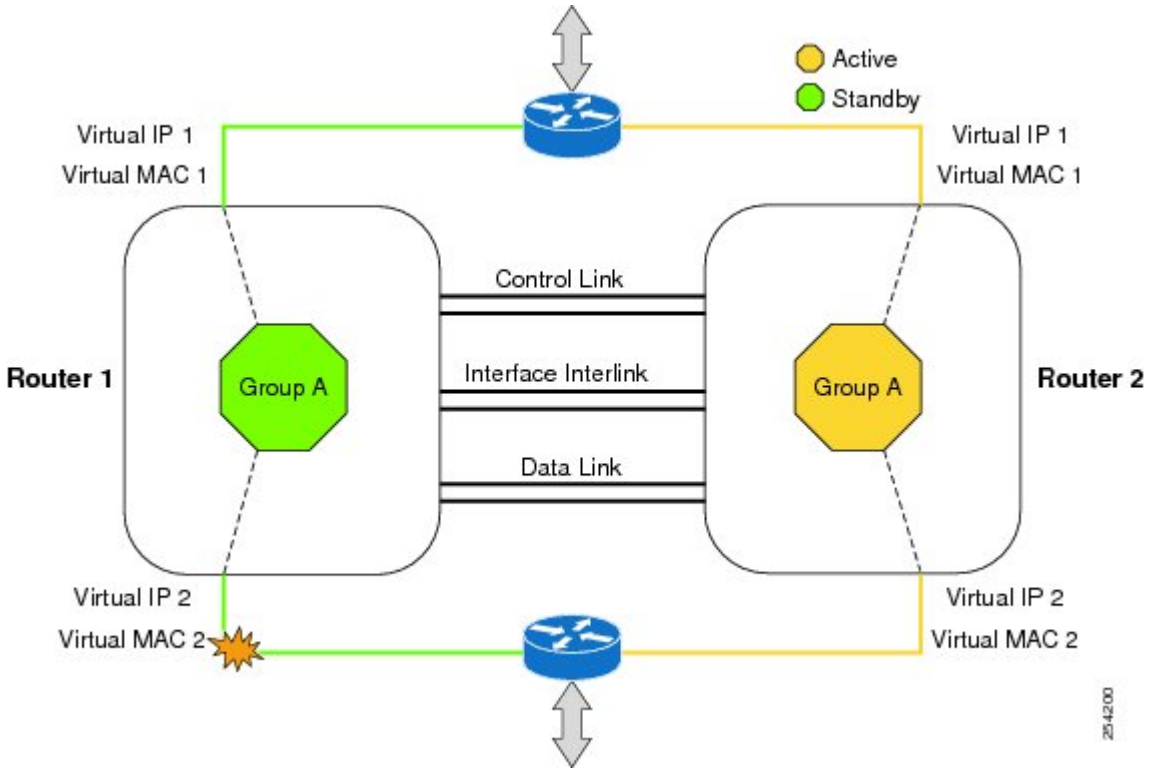
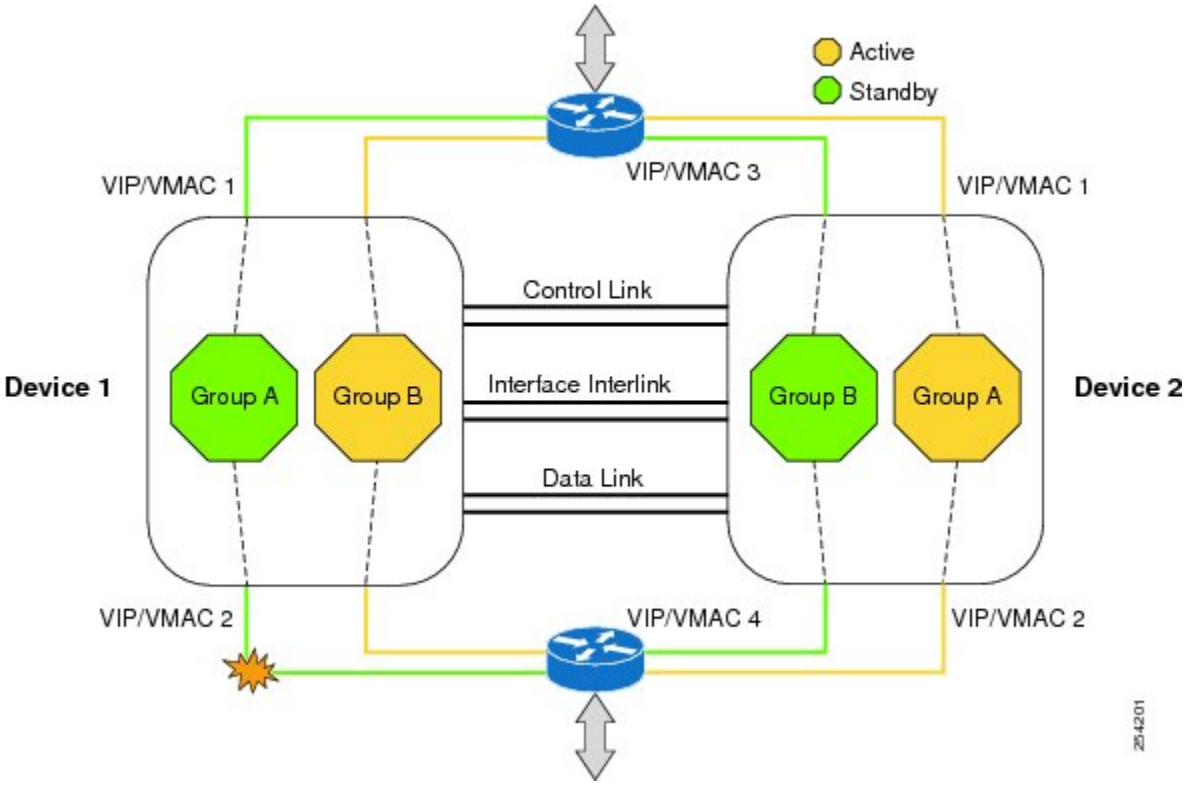


Figure 62: Redundancy Group Configuration--Two Outgoing Interfaces



The status of redundancy group members is determined through the use of hello messages sent over the control link. If either of the routers does not respond to a hello message within a configurable amount of time, it is considered that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for the hello messages:

- Active timer
- Standby timer
- Hellotime--The interval at which hello messages are sent
- Holdtime--The amount of time before the active or the standby router is declared to be down

The hellotime defaults to 3 seconds to align with Hot Standby Router Protocol (HSRP), and the holdtime defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine which pairs of interfaces are affected by the switchover, you must configure a unique ID number for each pair of redundant interfaces. This ID number is known as the RII associated with the interface.

A switchover to the standby router can also occur under other circumstances. Another factor that can cause a switchover is a priority setting that is configurable for each router. The router with the highest priority value will be the active router. If a fault occurs on either the active or the standby router, the priority of the router is decremented by a configurable amount known as the weight. If the priority of the active router falls below the priority of the standby router, a switchover occurs and the standby router becomes the active router. This default behavior can be overridden by disabling the preemption attribute for the redundancy group. You can

also configure each interface to decrease the priority when the L1 state of the interface goes down. This amount overrides the default amount configured for the redundancy group.

Each failure event that causes a modification of a redundancy group's priority generates a syslog entry that contains a time stamp, the redundancy group that was affected, previous priority, new priority, and a description of the failure event cause.

Another situation that will cause a switchover to occur is when the priority of a router or interface falls below a configurable threshold level.

In general, a switchover to the standby router occurs under the following circumstances:

- Power loss or reload occurs on the active router (this includes crashes).
- The run-time priority of the active router goes down below that of the standby router.
- The run-time priority of the active router goes down below the configured threshold value.
- The redundancy group on the active router is reloaded manually using the **redundancy application reload group** *rg-number* command.
- Two consecutive hello messages missed on any monitored interface forces the interface into testing mode. When this occurs, both units first verify the link status on the interface and then execute the following tests:
  - Network activity test
  - ARP test
  - Broadcast ping test

In the Firewall Stateful Inter-Chassis Redundancy feature, the redundancy group traffic is routed through the virtual IP address that is associated with the ingress interface of the redundancy group. The traffic sent to the virtual IP address is received by the router that has the redundancy group in the active state. During a redundancy group failover, the traffic to the virtual IP address is automatically routed to the newly active redundancy group.

The firewall drops the traffic that arrives on the standby redundancy group in case the redundancy group traffic is routed through the physical IP address of a standby router and the traffic reaches the standby redundancy group. However, when the traffic arrives on the active redundancy group, the established TCP or UDP sessions are synchronized to the standby redundancy group.

Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC. The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC. When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP and programs the interface's Ethernet controller to accept packets destined for the VMAC.

IPv6 Support

You can assign each redundancy group (RG) on a traffic interface for both IPv4 and IPv6 virtual IP (VIP) addresses under the same redundancy interface identifier (RII). Each RG uses a unique virtual MAC (VMAC) address per RII. For an RG, the IPv6 link-local VIP and global VIP coexist on an interface.

You can configure an IPv4 VIP, a link-local IPv6 VIP, and/or a global IPv6 VIP for each RG on a traffic interface. IPv6 link-local VIP is mainly used when configuring static or default routes, whereas IPv6 global VIP is widely used in both LAN and WAN topologies.

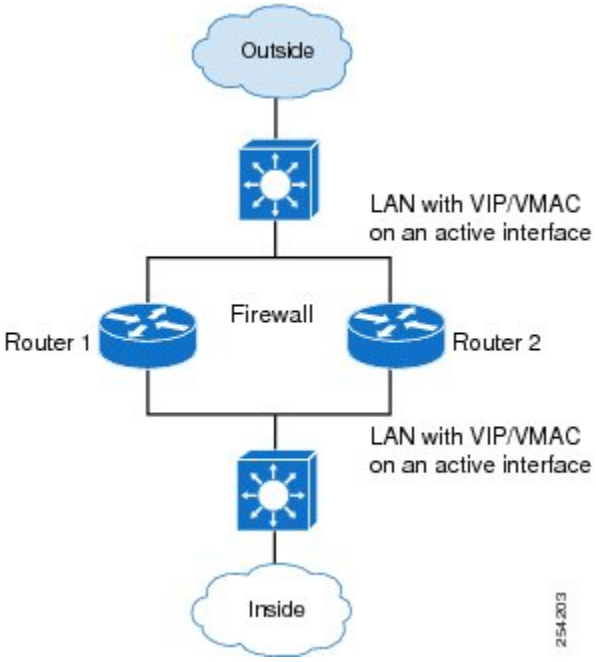
You must configure a physical IP address before configuring an IPv4 VIP.

Supported Topologies

The LAN-LAN topology is supported in the Firewall Stateful Inter-Chassis Redundancy architecture:

LAN-LAN

The figure below shows the LAN-LAN topology. When a dedicated appliance-based firewall solution is used, traffic is often directed to the correct firewall by configuring static routing in the upstream or downstream routers to an appropriate virtual IP address. In addition, the Aggregation Services Routers (ASRs) will participate in dynamic routing with upstream or downstream routers. The dynamic routing configuration supported on LAN facing interfaces must not introduce a dependency on routing protocol convergence; otherwise, fast failover requirements will not be met.



For more information about the LAN-LAN configuration, see the section, Example Configuring LAN-LAN.

VRF-Aware Interchassis Redundancy in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.

The VRF-Aware Interchassis Redundancy in Zone-Based Firewalls feature uses a VRF mapping mechanism that sends the VRF hash key along with box-to-box high availability session sync messages across active and standby devices.

How to Configure Firewall Stateful Interchassis Redundancy

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]
9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:**	Enters redundancy application configuration mode.

	Command or Action	Purpose
	`Device(config-red)# application redundancy`	
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.
Step 9	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:** `Device(config-red-app-grp)# track 200 decrement 200`	Specifies the priority value of a redundancy group that will be decremented if an event occurs.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Group Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*
7. **timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*}
8. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*}

9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name prot1`	(Optional) Configures the redundancy group (RG) with a name.
Step 7	**timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9`	Specifies the interval between when hello messages are sent and the time period before which a device is declared to be down.
Step 8	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies the authentication information.
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Virtual IP Address and a Redundant Interface Identifier

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII) for a redundancy group. • The range is from 1 to 65535.
Step 5	**redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20`	Associates an interface with a redundancy group and enables a virtual IP address.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Control Interface and a Data Interface

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the redundancy group.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/2 protocol 1`	Specifies the control interface that is used by the redundancy group. • This interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time that a redundancy group will take to delay role negotiations that start after a fault occurs or the system is reloaded.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Managing and Monitoring Firewall Stateful Inter-Chassis Redundancy

Use the following commands to manage and monitor the Firewall Stateful Inter-Chassis Redundancy feature.

SUMMARY STEPS

1. **enable**
2. **debug redundancy application group config** {**all** | **error** | **event** | **func**}
3. **debug redundancy application group faults** {**all** | **error** | **event** | **fault** | **func**}
4. **debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}
5. **debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**}
6. **debug redundancy application group rii** {**error** | **event**}
7. **debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**}
8. **debug redundancy application group vp** {**error** | **event**}
9. **show redundancy application group** [*group-id* | **all**]
10. **show redundancy application transport** {**client** | **group** [*group-id*]}
11. **show redundancy application control-interface group** [*group-id*]
12. **show redundancy application faults group** [*group-id*]
13. **show redundancy application protocol** {*protocol-id* | **group** [*group-id*]
14. **show redundancy application if-mgr group** [*group-id*]
15. **show redundancy application data-interface group** [*group-id*]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug redundancy application group config** {**all**	**error**	**event**	**func**} **Example:** `Device# debug redundancy application group config all`	Displays the redundancy group application configuration.
Step 3	**debug redundancy application group faults** {**all**	**error**	**event**	**fault**	**func**}	Displays the redundancy group application fault.

	Command or Action	Purpose
	Example: `Device# debug redundancy application group faults error`	
Step 4	**debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**} **Example:** `Device# debug redundancy application group media timer`	Displays the redundancy group application group media information.
Step 5	**debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**} **Example:** `Device# debug redundancy application group protocol peer`	Displays the redundancy group application group protocol information.
Step 6	**debug redundancy application group rii** {**error** | **event**} **Example:** `Device# debug redundancy application group rii event`	Displays the redundancy group application group RII information.
Step 7	**debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**} **Example:** `Device# debug redundancy application group transport trace`	Displays the redundancy group application group transport information.
Step 8	**debug redundancy application group vp** {**error** | **event**} **Example:** `Device# debug redundancy application group vp event`	Displays the redundancy group application group VP information.
Step 9	**show redundancy application group** [*group-id* | **all**] **Example:** `Device# show redundancy application group all`	Displays the redundancy group information.
Step 10	**show redundancy application transport** {**client** | **group** [*group-id*]} **Example:** `Device# show redundancy application transport group 1`	Displays transport specific information for a redundancy group.

	Command or Action	Purpose
Step 11	**show redundancy application control-interface group** [*group-id*] **Example:** `Device# show redundancy application control-interface group 2`	Displays control interface information for a redundancy group.
Step 12	**show redundancy application faults group** [*group-id*] **Example:** `Device# show redundancy application faults group 2`	Displays fault-specific information for a redundancy group.
Step 13	**show redundancy application protocol** {*protocol-id* | **group** [*group-id*] **Example:** `Device# show redundancy application protocol 3`	Displays protocol specific information for a redundancy group.
Step 14	**show redundancy application if-mgr group** [*group-id*] **Example:** `Device# show redundancy application if-mgr group 2`	Displays interface manager information for a redundancy group.
Step 15	**show redundancy application data-interface group** [*group-id*] **Example:** `Device# show redundancy application data-interface group 1`	Displays data interface specific information.
Step 16	**end** **Example:** `Device# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for Firewall Stateful Interchassis Redundancy

Example: Configuring a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover-threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 200 decrement 200
Device(config-red-app-grp)# end
```

Example: Configuring a Redundancy Group Protocol

The following example shows how to configure a redundancy group with timers set for hello time and hold time messages:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring a Virtual IP Address and a Redundant Interface Identifier

The following example shows how to configure the redundancy group virtual IP address for Gigabit Ethernet interface 0/1/1:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/1
Device(conf-if)# redundancy rii 600
Device(config-if)# redundancy group 2 ip 10.2.3.4 exclusive decrement 200
Device(config-if)# end
```

Example: Configuring a Control Interface and a Data Interface

```
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# end
```

Example: Configuring a LAN-LAN Topology

The following is a sample LAN-LAN configuration that shows how a pair of routers that have two outgoing interfaces are configured for stateful redundancy. In this example, GigabitEthernet 0/1/1 is the ingress interface and GigabitEthernet 0/2/1 is the egress interface. Both interfaces are assigned to zones and a classmap is defined to describe the traffic between zones. Interfaces are also configured for redundancy. The “inspect” action invokes the application-level gateway (ALG) to open a pinhole to allow traffic on other ports. A pinhole is a port that is opened through an ALG to allow a particular application to gain controlled access to a protected network.

The following is the configuration on Device 1, the active device.

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.3 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.10 exclusive decrement 50
```

```
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.2 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.17 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.49 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

The following is the configuration on Device 2, the standby device:

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
```

```
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.6 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.5 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.10 exclusive decrement 50
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.21 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.53 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Stateful Interchassis Redundancy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 173: Feature Information for Firewall Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Firewall Stateful Interchassis Redundancy	Cisco IOS XE Release 3.1(S)	The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act a backups for each other. The following commands were introduced or modified: **application redundancy**, **authentication**, **control**, **data**, **debug redundancy application group config**, **debug redundancy application group faults**, **debug redundancy application group media**, **debug redundancy application group protocol**, **debug redundancy application group rii**, **debug redundancy application group transport**, **debug redundancy application group vp**, **group**, **name**, **preempt**, **priority**, **protocol**, **redundancy rii**, **redundancy group**, **track**, **timers delay**, **timers hellotime**, **show redundancy application group**, **show redundancy application transport**, **show redundancy application control-interface**, **show redundancy application faults**, **show redundancy application protocol**, **show redundancy application if-mgr**, **show redundancy application data-interface**.

Feature Name	Releases	Feature Information
VRF-Aware Stateful Interchassis Redundancy in Zone-Based Firewalls	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.

CHAPTER 129

Firewall Box to Box High Availability Support for Cisco CSR1000v Routers

The Firewall Box to Box High Availability Support on Cisco CSR1000v Routers feature enables you to configure pairs of routers to act as backup for each other. This feature can be configured to determine the active router based on a number of failover conditions. When a failover occurs, the standby router seamlessly takes over and starts performing traffic forwarding services and maintaining a dynamic routing table.

Prerequisites for Firewall Box-to-Box High Availability Support for Cisco CSR1000v Routers

- The interfaces attached to the firewall must have the same redundant interface identifier (RII).
- The active device and the standby device must have the same Cisco IOS XE Zone-Based Firewall configuration.
- The active device and the standby device must run on an identical version of the Cisco IOS XE software. The active device and the standby device must be connected through a switch.

Restrictions for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall box-to-box high availability configuration.

Information About Firewall Box to Box High Availability Support on Cisco CSR1000v Routers

How Firewall Box to Box High Availability Support on Cisco CSR1000v Works

You can configure pairs of routers to act as hot standbys for each other. This redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups. The figure below depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that has one outgoing interface. The Redundancy Group Configuration—Two Outgoing Interfaces figure depicts the active-active device scenario shows how two redundancy groups are configured for a pair of routers that have two outgoing interfaces.

Note that in both cases, the redundant routers are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and to synchronize the stateful database for these applications.

Also, in both cases, the pairs of redundant interfaces are configured with the same unique ID number known as the RII.

Figure 63: Redundancy Group Configuration—Two Outgoing Interfaces

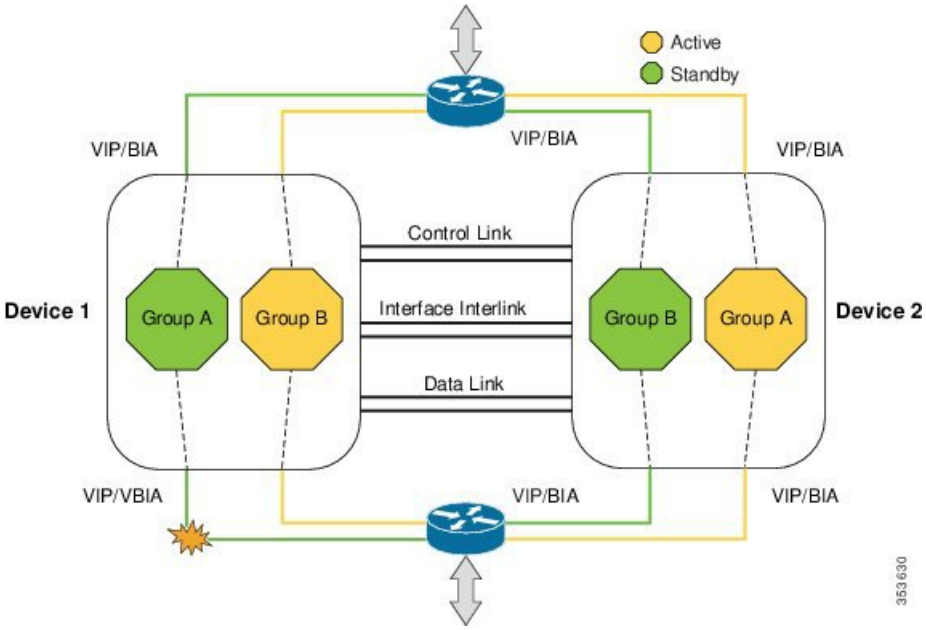
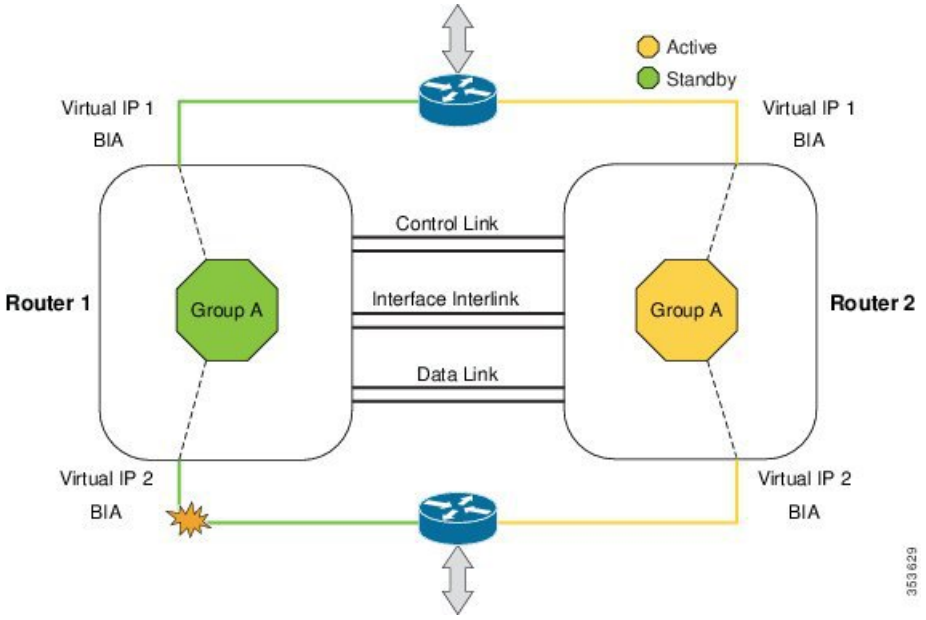
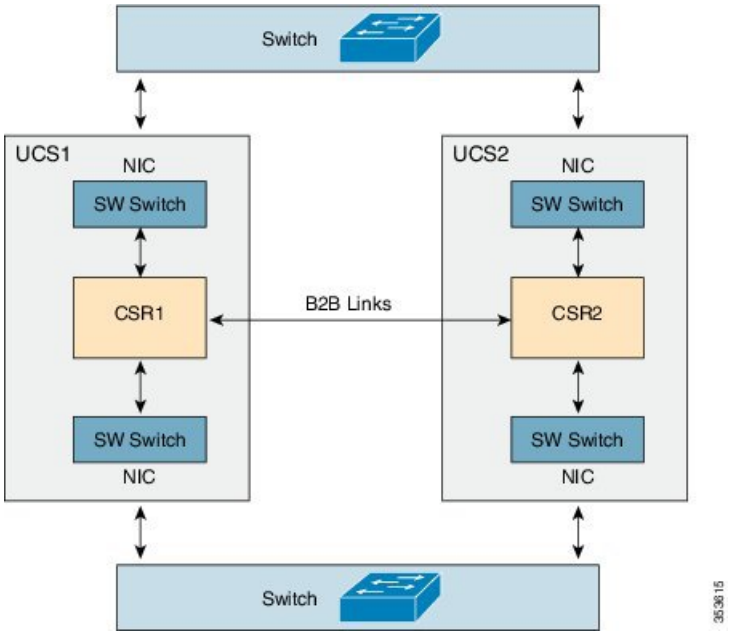


Figure 64: Redundancy Group Configuration



The following scenarios are examples of Box-to-Box High Availability deployment for Cisco CSR1000v routers:

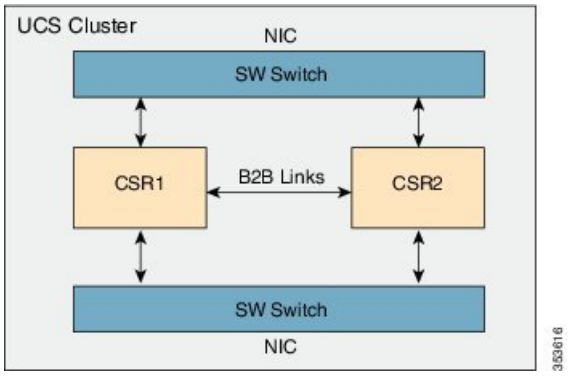
Figure 65: CSR1000v Box-to-Box High Availability on Two Independent Servers



In this deployment, two redundant Cisco CSR 1000v routers are in two independent UCS servers. The two Cisco Unified Computing System (UCS) servers can be in the same data center or two different data centers in different regions. We recommended that you configure two individual physical connections for box-to-box high availability data and control links. However, if the two dedicated physical links are not available, the box-to-box high availability data and control traffic can go through different LAN extension connections. Box-to-Box high availability parameters, such as heart beat period need to be adjusted to take into account the extended delay.

LAN interfaces of each Cisco CSR 1000v router are connected with UCS physical network interface card (NIC) interfaces through switches (for example, ESXi L2 SW). The two physical NICs on each UCS are connected to outside switch to form a box-to-box pair. Gratuitous Address Resolution Protocols (ARP) is sent from CSR LAN interfaces to reach physical switch and its Built-in Address (BIA).

Figure 66: CSR1000v Box-to-Box High Availability on Cluster Server



In the above deployment, NAT and Zone-Based Firewall (ZBFW) box-to-box high availability also works on UCS cluster setup. In this case, box-to-box control and data links go through virtual connections within the cluster. Switches (For example, ESXi L2 SW) are used to connect the 2 redundant Cisco CSR 1000v

routers to form a box-to-box high availability pair; LAN interfaces on two Cisco CSR 1000v routers are connected directly to the SW switches, and two physical NICs of the cluster UCS are connected with the SW switches to communicate outside the network.

Refer to the *Configuring Firewall Stateful Interchassis Redundancy* module for additional information on configurations and examples.

Configuration Example for Firewall Box-to-Box High Availability Support for Cisco CSR 1000v Routers

Example: Configuring Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

The following examples shows how to configure a redundancy application group, a redundancy group protocol, Virtual IP Address and Redundant Interface Identifier, and control and data interfaces:

```
!Configures a redundancy application group
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover-threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 200 decrement 200
Device(config-red-app-grp)# exit

!Configures a redundancy group protocol
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end

! Configures a Virtual IP Address and Redundant Interface Identifier
Device# configure terminal
Device(config)# interface GigabitEthernet0/1/1
Device(conf-if)# redundancy rii 600
Device(config-if)# redundancy group 2 ip 10.2.3.4 exclusive decrement 200
Device(config)# redundancy
Device(config-red-app-grp)# data GigabitEthernet0/0/0
Device(config-red-app-grp)# control GigabitEthernet0/0/2 protocol 1
Device(config-red-app-grp)# end

!Configures control and data interfaces
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# end
```

Additional References for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Firewall Stateful Interchassis Redundancy	• Configuring Firewall Stateful Interchassis Redundancy

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Box-to-Box High Availability for Cisco CSR1000v Routers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 174: Feature Information for Firewall Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Firewall Box-to-Box High Availability for Cisco CSR1000v Routers	Cisco IOS XE Release 3.14S	The Firewall Box-to-Box High Availability for Cisco CSR1000v Routers feature enables you to configure pairs of Cisco CSR1000v routers to act a backups for each other.

CHAPTER 130

Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the router that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session.

This module provides an overview of asymmetric routing and describes how to configure asymmetric routing

Restrictions for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The following restrictions apply to the Interchassis Asymmetric Routing Support feature:

- LANs that use virtual IP addresses and virtual MAC (VMAC) addresses do not support asymmetric routing.
- In Service Software Upgrade (ISSU) is not supported.

The following features are not supported by the VRF-Aware Asymmetric Routing Support feature:

- Cisco Trustsec
- Edge switching services
- Header compression

- IPsec
- Policy Based Routing (PBR)
- Port bundle
- Lawful intercept
- Layer 2 Tunneling Protocol (L2TP)
- Locator/ID Separation Protocol (LISP) inner packet inspection
- Secure Shell (SSL) VPN
- Session Border Controller (SBC)

Information About Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

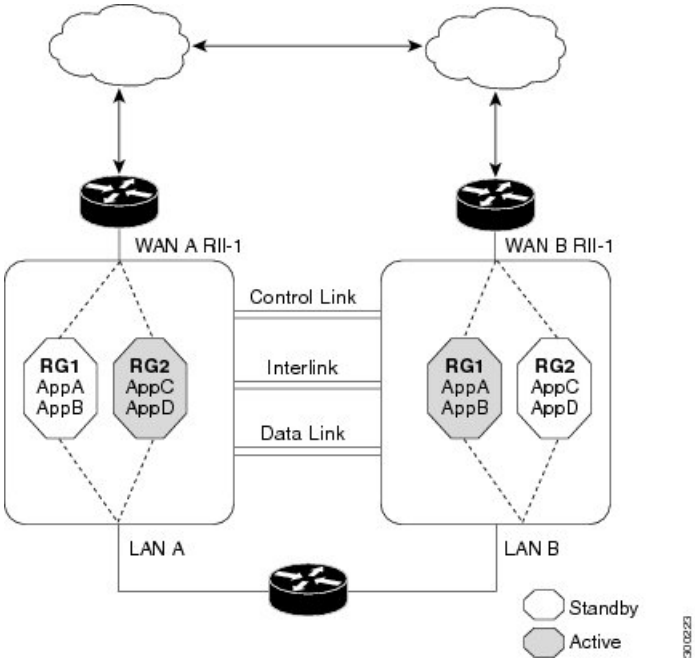
Asymmetric Routing Overview

Asymmetric routing occurs when packets from TCP or UDP connections flow in different directions through different routes. In asymmetric routing, packets that belong to a single TCP or UDP connection are forwarded through one interface in a redundancy group (RG), but returned through another interface in the same RG. In asymmetric routing, the packet flow remains in the same RG. When you configure asymmetric routing, packets received on the standby RG are redirected to the active RG for processing. If asymmetric routing is not configured, the packets received on the standby RG may be dropped.

Asymmetric routing determines the RG for a particular traffic flow. The state of the RG is critical in determining the handling of packets. If an RG is active, normal packet processing is performed. In case the RG is in a standby state and you have configured asymmetric routing and the **asymmetric-routing always-divert enable** command, packets are diverted to the active RG. Use the **asymmetric-routing always-divert enable** command to always divert packets received from the standby RG to the active RG.

The figure below shows an asymmetric routing scenario with a separate asymmetric-routing interlink interface to divert packets to the active RG.

Figure 67: Asymmetric Routing Scenario



The following rules apply to asymmetric routing:

- 1:1 mapping exists between the redundancy interface identifier (RII) and the interface.
- 1:*n* mapping exists between the interface and an RG. (An asymmetric routing interface can receive traffic from and send traffic to multiple RGs. For a non asymmetric-routing interface (normal LAN interface), a 1:1 mapping exists between the interface and the RG.)
- 1:*n* mapping exists between an RG and applications that use it. (Multiple applications can use the same RG).
- 1:1 mapping exists between an RG and the traffic flow. The traffic flow must map only to a single RG. If a traffic flow maps to multiple RGs, an error occurs.
- 1:1 or 1:*n* mapping can exist between an RG and an asymmetric-routing interlink as long as the interlink has sufficient bandwidth to support all the RG interlink traffic.

Asymmetric routing consists of an interlink interface that handles all traffic that is to be diverted. The bandwidth of the asymmetric-routing interlink interface must be large enough to handle all expected traffic that is to be diverted. An IPv4 address must be configured on the asymmetric-routing interlink interface, and the IP address of the asymmetric routing interface must be reachable from this interface.

Note We recommend that the asymmetric-routing interlink interface be used for interlink traffic only and not be shared with high availability control or data interfaces because the amount of traffic on the asymmetric-routing interlink interface could be quite high.

Asymmetric Routing Support in Firewalls

For intrabox asymmetric routing support, the firewall does a stateful Layer 3 and Layer 4 inspection of Internet Control Message Protocol (ICMP), TCP, and UDP packets. The firewall does a stateful inspection of TCP packets by verifying the window size and order of packets. The firewall also requires the state information from both directions of the traffic for stateful inspection. The firewall does a limited inspection of ICMP information flows. It verifies the sequence number associated with the ICMP echo request and response. The firewall does not synchronize any packet flows to the standby redundancy group (RG) until a session is established for that packet. An established session is a three-way handshake for TCP, the second packet for UDP, and informational messages for ICMP. All ICMP flows are sent to the active RG.

The firewall does a stateless verification of policies for packets that do not belong to the ICMP, TCP, and UDP protocols.

The firewall depends on bidirectional traffic to determine when a packet flow should be aged out and diverts all inspected packet flows to the active RG. Packet flows that have a pass policy and that include the same zone with no policy or a drop policy are not diverted.

Note The firewall does not support the **asymmetric-routing always-divert enable** command that diverts packets received on the standby RG to the active RG. By default, the firewall forces all packet flows to be diverted to the active RG.

Asymmetric Routing in NAT

By default, when asymmetric routing is configured, Network Address Translation (NAT) processes non-ALG packets on the standby RG, instead of forwarding them to the active. The NAT-only configuration (that is when the firewall is not configured) can use both the active and standby RGs for processing packets. If you have a NAT-only configuration and you have configured asymmetric routing, the default asymmetric routing rule is that NAT will selectively process packets on the standby RG. You can configure the **asymmetric-routing always-divert enable** command to divert packets received on the standby RG to the active RG. Alternatively, if you have configured the firewall along with NAT, the default asymmetric routing rule is to always divert the packets to the active RG.

When NAT receives a packet on the standby RG and if you have not configured the diverting of packets, NAT does a lookup to see if a session exists for that packet. If a session exists and there is no ALG associated for that session, NAT processes the packet on the standby RG. The processing of packets on the standby RG when a session exists significantly increases the bandwidth of the NAT traffic.

ALGs are used by NAT to identify and translate payload and to create child flows. ALGs require a two-way traffic to function correctly. NAT must divert all traffic to the active RG for any packet flow that is associated with an ALG. This is accomplished by checking if ALG data that is associated with the session is found on the standby RG. If ALG data exits, the packet is diverted for asymmetric routing.

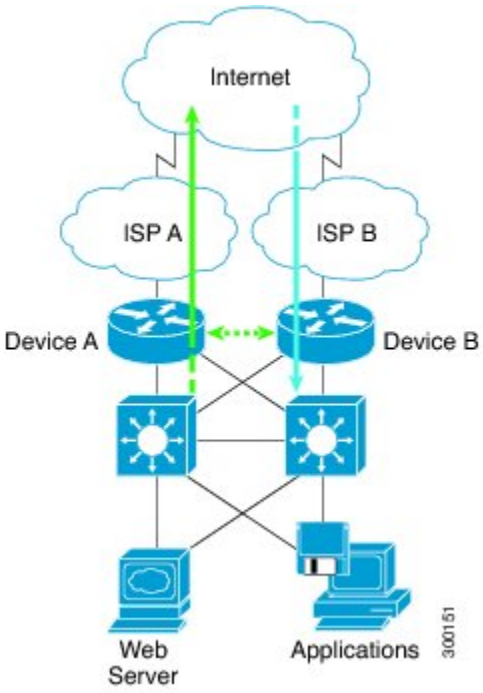
VRF-Aware Software Infrastructure (VASI) support was added in Cisco IOS XE Release 3.16S. Multiprotocol Label Switching (MPLS) asymmetric routing is also supported.

In Cisco IOS XE Release 3.16S, NAT supports asymmetric routing with ALGs, Carrier Grade NAT (CGN), and virtual routing and forwarding (VRF) instances. No configuration changes are required to enable asymmetric routing with ALGs, CGN, or VRF. For more information, see the section, “Example: Configuring Asymmetric Routing with VRF”.

Asymmetric Routing in a WAN-LAN Topology

Asymmetric routing supports only a WAN-LAN topology. In a WAN-LAN topology, devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links. Asymmetric routing controls the routing of return traffic received through WAN links in a WAN-LAN topology. The figure below shows a WAN-LAN topology.

Figure 68: Asymmetric Routing in a WAN-LAN Topology



VRF-Aware Asymmetric Routing in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. The feature supports Multiprotocol Label Switching (MPLS).

During asymmetric routing diversion, the VPN routing and forwarding (VRF) name hash value is sent with diverted packets. The VRF name hash value is converted to the local VRF ID and table ID at the active device after the diversion.

When diverted packets reach the active device on which Network Address Translation (NAT) and the zone-based firewall are configured, the firewall retrieves the VRF ID from NAT or NAT64 and saves the VRF ID in the firewall session key.

The following section describes the asymmetric routing packet flow when only the zone-based firewall is configured on a device:

- When MPLS is configured on a device, the VRF ID handling for diverted packets is the same as the handling of non-asymmetric routing diverted packets. An MPLS packet is diverted to the active device, even though the MPLS label is removed at the standby device. The zone-based firewall inspects the packet at the egress interface, and the egress VRF ID is set to zero, if MPLS is detected at this interface. The firewall sets the ingress VRF ID to zero if MPLS is configured at the ingress interface.

- When a Multiprotocol Label Switching (MPLS) packet is diverted to the active device from the standby device, the MPLS label is removed before the asymmetric routing diversion happens.
- When MPLS is not configured on a device, an IP packet is diverted to the active device and the VRF ID is set. The firewall gets the local VRF ID, when it inspects the packet at the egress interface.

VRF mapping between active and standby devices require no configuration changes.

VRF-Aware Asymmetric Routing in NAT

In Cisco IOS XE Release 3.14S, Network Address Translation supports VRF-aware interchassis asymmetric routing. VRF-aware interchassis asymmetric routing uses message digest (MD) 5 hash of the VPN routing and forwarding (VRF) name to identify the VRF and datapath in the active and standby devices to retrieve the local VRF ID from the VRF name hash and viceversa.

For VRF-aware interchassis asymmetric routing, the VRFs on active and standby devices must have the same VRF name. However, the VRF ID need not be identical on both devices because the VRF ID is mapped based on the VRF name on the standby and active devices during asymmetric routing diversion or box-to-box high availability synchronization.

In case of MD5 hash collision for VRF names, the firewall and NAT sessions that belong to the VRF are not synced to the standby device.

VRF mapping between active and standby devices require no configuration changes.

How to Configure Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Configuring a Redundancy Application Group and a Redundancy Group Protocol

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.
- Faults (objects) that decrement the priority
- Failover priority
- Failover threshold
- Group instance
- Group name
- Initialization delay timer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**

4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **priority** *value* [**failover threshold** *value*]
8. **preempt**
9. **track** *object-number* **decrement** *number*
10. **exit**
11. **protocol** *id*
12. **timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*}
13. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*}
14. **bfd**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	Specifies an optional alias for the protocol instance.
Step 7	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	Specifies the initial priority and failover threshold for a redundancy group.

	Command or Action	Purpose
Step 8	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the redundancy group and enables the standby device to preempt the active device. • The standby device preempts only when its priority is higher than that of the active device.
Step 9	**track** *object-number* **decrement** *number* **Example:** `Device(config-red-app-grp)# track 50 decrement 50`	Specifies the priority value of a redundancy group that will be decremented if an event occurs on the tracked object.
Step 10	**exit** **Example:** `Device(config-red-app-grp)# exit`	Exits redundancy application group configuration mode and enters redundancy application configuration mode.
Step 11	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 12	**timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10`	Specifies the interval between hello messages sent and the time period before which a device is declared to be down. • Holdtime should be at least three times the hellotime.
Step 13	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies authentication information.
Step 14	**bfd** **Example:** `Device(config-red-app-prtcl)# bfd`	Enables the integration of the failover protocol running on the control interface with the Bidirectional Forwarding Detection (BFD) protocol to achieve failure detection in milliseconds. • BFD is enabled by default.
Step 15	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring Data, Control, and Asymmetric Routing Interfaces

In this task, you configure the following redundancy group (RG) elements:

- The interface that is used as the control interface.

- The interface that is used as the data interface.
- The interface that is used for asymmetric routing. This is an optional task. Perform this task only if you are configuring asymmetric routing for Network Address Translation (NAT).

Note Asymmetric routing, data, and control must be configured on separate interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **asymmetric-routing interface** *type number*
10. **asymmetric-routing always-divert enable**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group (RG) and enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/1`	Specifies the data interface that is used by the RG.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1`	Specifies the control interface that is used by the RG. • The control interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time required for an RG to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**asymmetric-routing interface** *type number* **Example:** `Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1`	Specifies the asymmetric routing interface that is used by the RG.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Always diverts packets received from the standby RG to the active RG.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

Note
- You must not configure a redundant interface identifier (RII) on an interface that is configured either as a data interface or as a control interface.
- You must configure the RII and asymmetric routing on both active and standby devices.
- You cannot enable asymmetric routing on the interface that has a virtual IP address configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **redundancy rii** *id*
5. **redundancy group** *id* [**decrement** *number*]
6. **redundancy asymmetric-routing enable**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/3`	Selects an interface to be associated with the redundancy group (RG) and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII).
Step 5	**redundancy group** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 decrement 20`	Enables the RG redundancy traffic interface configuration and specifies the amount to be decremented from the priority when the interface goes down. **Note** You need not configure an RG on the traffic interface on which asymmetric routing is enabled.
Step 6	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Establishes an asymmetric flow diversion tunnel for each RG.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring Dynamic Inside Source Translation with Asymmetric Routing

The following configuration is a sample dynamic inside source translation with asymmetric routing. You can configure asymmetric routing with the following types of NAT configurations—dynamic outside source, static inside and outside source, and Port Address Translation (PAT) inside and outside source translations.

For more information on different types of NAT configurations, see the "Configuring NAT for IP Address Conservation" chapter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **exit**
7. **redundancy**
8. **application redundancy**
9. **group** *id*
10. **asymmetric-routing always-divert enable**
11. **end**
12. **configure terminal**
13. **ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*}
14. **exit**
15. **ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id*
16. **access-list** *standard-acl-number* **permit** *source-address wildcard-bits*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/3`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**redundancy** **Example:** `Device(config)# redundancy`	Configures redundancy and enters redundancy configuration mode.
Step 8	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 9	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Diverts the traffic to the active device.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.
Step 12	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 13	**ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool pool1 prefix-length 24`	Defines a pool of global addresses. • Enters IP NAT pool configuration mode.
Step 14	**exit** **Example:** `Device(config-ipnat-pool)# exit`	Exits IP NAT pool configuration mode and enters global configuration mode.
Step 15	**ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id* **Example:** `Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100`	Enables NAT of the inside source address and associates NAT with a redundancy group by using the mapping ID.

	Command or Action	Purpose
Step 16	**access-list** *standard-acl-number* **permit** *source-address wildcard-bits* **Example:** `Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0`	Defines a standard access list for the inside addresses that are to be translated.
Step 17	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Example: Configuring a Redundancy Application Group and a Redundancy Group Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 50 decrement 50
Device(config-red-app-grp)# exit
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring Data, Control, and Asymmetric Routing Interfaces

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/1
Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
```

Example: Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/3
Device(config-if)# redundancy rii 600
Device(config-if)# redundancy group 1 decrement 20
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Example: Configuring Dynamic Inside Source Translation with Asymmetric Routing

```
Device(config)# interface gigabitethernet 0/1/3
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# ip nat pool pool1 prefix-length 24
Device(config-ipnat-pool)# exit
Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100
Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0
```

Example: Configuring VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The following is a sample WAN-to-WAN symmetric routing configuration:

```
vrf definition Mgmt-intf
 address-family ipv4
  exit-address-family
!
address-family ipv6
 exit-address-family
!
!
vrf definition VRFA
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 address-family ipv4
  exit-address-family
!
!
no logging console
no aaa new-model
```

```
!
multilink bundle-name authenticated
!
redundancy
 mode sso
 application redundancy
  group 1
   preempt
   priority 120
   control GigabitEthernet 0/0/1 protocol 1
   data GigabitEthernet 0/0/2
!
!
!
!
ip tftp source-interface GigabitEthernet0
ip tftp blocksize 8192
!
track 1 interface GigabitEthernet 0/0/4 line-protocol
!
interface Loopback 0
 ip address 209.165.201.1 255.255.255.224
!
interface GigabitEthernet 0/0/0
 vrf forwarding VRFA
 ip address 192.168.0.1 255.255.255.248
 ip nat inside
 negotiation auto
 bfd interval 50 min_rx 50 multiplier 3
 redundancy rii 2
!
interface GigabitEthernet 0/0/1
 ip address 209.165.202.129 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/2
 ip address 192.0.2.1 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/3
 ip address 198.51.100.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0
 vrf forwarding Mgmt-intf
 ip address 172.16.0.1 255.255.0.0
 negotiation auto
!
interface vasileft 1
 vrf forwarding VRFA
 ip address 10.4.4.1 255.255.0.0
 ip nat outside
 no keepalive
!
interface vasiright 1
ip address 10.4.4.2 255.255.0.0
no keepalive
!
router mobile
!
```

```
router bgp 577
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor 203.0.113.1 remote-as 223
 neighbor 203.0.113.1 description PEERING to PTNR neighbor 10.4.4.1 remote-as 577
 neighbor 10.4.4.1 description PEEERING to VASI VRFA interface
!
address-family ipv4
 network 203.0.113.1 mask 255.255.255.240
 network 10.4.0.0 mask 255.255.0.0
 network 209.165.200.224 mask 255.255.255.224
 neighbor 203.0.113.1 activate
 neighbor 10.4.4.1 activate
 neighbor 10.4.4.1 next-hop-self
 exit-address-family
!
address-family ipv4 vrf VRFA
 bgp router-id 4.4.4.4
 network 192.168.0.0 mask 255.255.255.248
 network 10.4.0.0 mask 255.255.0.0
 redistribute connected
 redistribute static
 neighbor 192.168.0.2 remote-as 65004
 neighbor 192.168.0.2 fall-over bfd
 neighbor 192.168.0.2 activate
 neighbor 10.4.4.2 remote-as 577
 neighbor 10.4.4.2 description PEERING to VASI Global intf
 neighbor 10.4.4.2 activate
 exit-address-family
!
ip nat switchover replication http
ip nat pool att_pool 209.165.200.225 209.165.200.225 prefix-length 16
ip nat inside source list 4 pool att_pool redundancy 1 mapping-id 100 vrf VRFA overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip route 203.0.113.1 255.255.255.224 10.4.4.1
ip route 192.168.0.0 255.255.0.0 10.4.4.1
ip route 209.165.200.224 255.255.255.224 10.4.4.1
ip route vrf Mgmt-intf 209.165.200.1 255.255.255.224 172.16.0.0
!
ip prefix-list VRF_Pool seq 5 permit 209.165.200.0/27
ip prefix-list p1-adv-1 seq 5 permit 209.165.200.0/27
ip prefix-list p1-exist-1 seq 5 permit 203.0.113.193/27
logging esm config
access-list 4 permit 203.0.113.193 255.255.255.224
!
control-plane
line console 0
 stopbits 1
!
line vty 0 3
 login
!
line vty 4
 password lab
 login
!
end
```

Example: Configuring Asymmetric Routing with VRF

The following example shows how to configure asymmetric routing with virtual routing and forwarding (VRF) instances:

```
Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# priority 100 failover threshold 40
Device(config-red-app-grp)# control GigabitEthernet 1/0/3 protocol 1
Device(config-red-app-grp)# data GigabitEthernet 1/0/3
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 1/0/4
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# exit
Device(config-red-app)# exit
Device(config-red)# exit
!
Device(config)# interface TenGigabitEthernet 2/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit
!
Device(config)# interface TenGigabitEthernet 3/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
!
Device(config-if)# ip nat pool pool-vrf001 209.165.201.1 209.165.201.30 prefix-length 24
Device(config-if)# ip nat inside source list 1 pool pool-vrf001 redundancy 1 mapping-id 1
vrf vrf001 match-in-vrf overload
Device(config-if)# end
```

Additional References for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Firewall inter-chassis redundancy	"Configuring Firewall Stateful Inter-Chassis Redundancy" module
NAT inter-chassis redundancy	"Configuring Stateful Inter-Chassis Redundancy" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 175: Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Feature Name	Releases	Feature Information
Asymmetric Routing Enhancements for NAT44	Cisco IOS XE Release 3.16S	The Asymmetric Routing Enhancements for NAT44 feature supports asymmetric routing with CGN, ALGs, VRF, VASI and MPLS. No commands were introduced or modified.
Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT	Cisco IOS XE Release 3.5S	The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. The following commands were introduced or modified: **asymmetric-routing**, **redundancy asymmetric-routing enable**.
VRF-Aware Interchassis Asymmetric Routing Support for Zone-Based Firewalls	Cisco IOS XE Release 3.14S	Zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.
VRF-Aware Interchassis Asymmetric Routing Support for NAT	Cisco IOS XE Release 3.14S	NAT supports the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.

CHAPTER 131

Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

The Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls feature supports high availability (HA) based on redundancy groups (RGs) on IPv6 firewalls. This feature enables you to configure pairs of devices to act as backup for each other. This feature can be configured to determine the active device based on a number of failover conditions. This feature supports the FTP66 application-layer gateway (ALG) for IPv6 packet inspection.

This module provides information about Box-to-Box (B2B) HA support and describes how to configure this feature.

Prerequisites for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

- Interfaces attached to a firewall must have the same redundant interface identifier (RII).
- Active and standby devices must have the same zone-based policy firewall configuration.
- Active and standby devices must run on identical versions of Cisco software. The active and standby devices must be connected through a switch.
- The box-to-box (B2B) configuration on both active and standby devices should be the same because there is no autosynchronization of the configuration between these devices.

- For asymmetric routing traffic to pass, you must configure the pass action for the class-default class. Class-default class is a system-defined class map that represents all packets that do not match any of the user-defined classes in a policy.
- If you configure a zone pair between two LAN interfaces, ensure that you configure the same redundancy group (RG) on both interfaces. The zone pair configuration is not supported if LAN interfaces belong to different RGs.

Restrictions for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

- Only IPv4 is supported at box-to-box (B2B) interlink interfaces.
- Multiprotocol Label Switching (MPLS) and virtual routing and forwarding (VRF) are not supported.
- Cisco ASR 1006 and 1013 Aggregation Services Routers with dual Embedded Services Processors (ESPs) or dual Route Processors (RPs) in the chassis are not supported, because coexistence of interbox high availability (HA) and intrabox HA is not supported.

  Cisco ASR 1006 and Cisco ASR 1013 Aggregation Services Routers with single ESP and single RP in the chassis support interchassis redundancy.
- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall stateful interchassis redundancy configuration.
- Stateless Network Address Translation 64 (NAT64) with IPv6 firewalls is not supported.

Information About Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Zone-Based Policy Firewall High Availability Overview

High availability enables network-wide protection by providing fast recovery from faults that may occur in any part of a network. High availability enables rapid recovery from disruptions to users and network applications.

The zone-based policy firewall supports active/active and active/standby high availability failover and asymmetric routing.

The active/active failover allows both devices involved in the failover to forward traffic simultaneously.

When active/standby high availability failover is configured, only one of the devices involved in the failover handles the traffic at one time, while the other device is in a standby mode, periodically synchronizing session information from the active device.

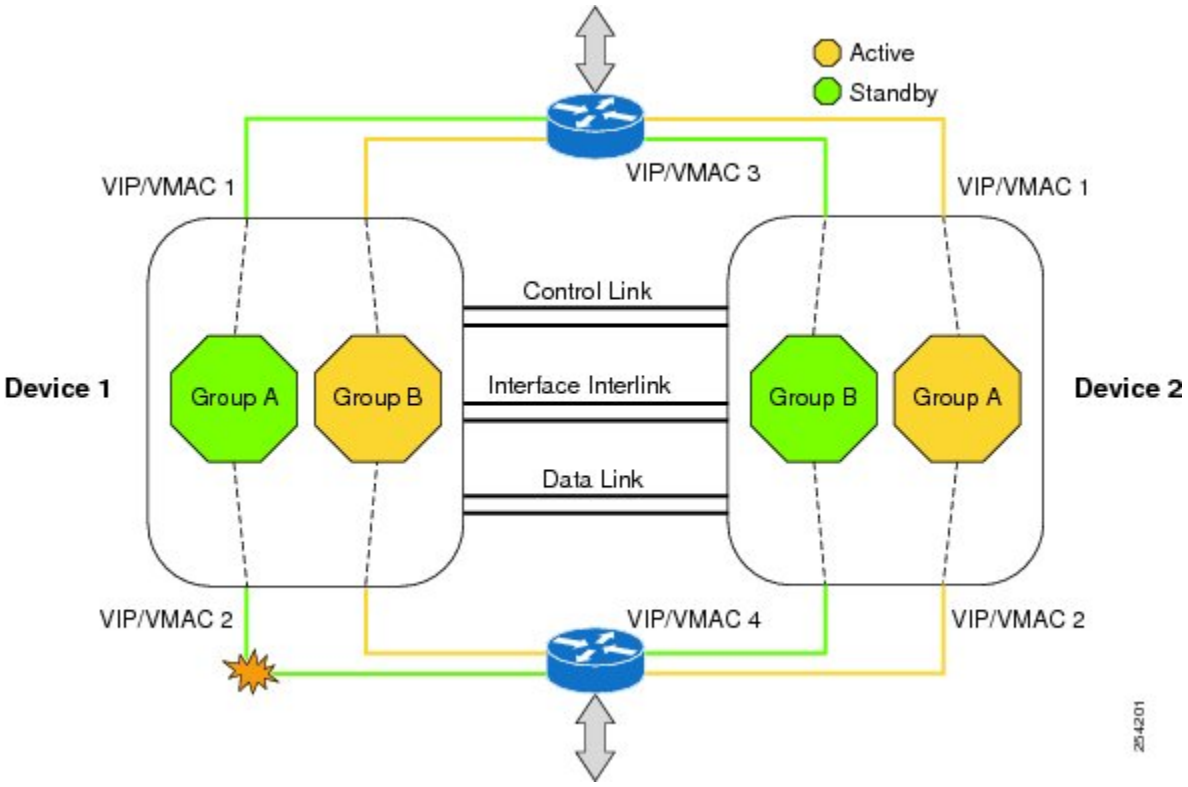
Asymmetric routing supports the forwarding of packets from a standby redundancy group to an active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the

device that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session.

Box-to-Box High Availability Operation

You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured per interface. Pairs of redundant interfaces are known as redundancy groups (RGs). Figure 1 depicts an active/active failover scenario. It shows how two redundancy groups are configured for a pair of devices that have two outgoing interfaces.

Figure 69: Redundancy Group Configuration—Two Outgoing Interfaces



The redundant devices are joined by a configurable control link, a data synchronization link, and an interlink interface. The control link is used to communicate the status of the devices. The data synchronization link is used to transfer stateful information from the firewall and to synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number, known as the redundant interface identifier (RII). The routing table is not synced from active to standby.

Asymmetric routing is supported as part of the firewall HA. In a LAN-WAN scenario, where the return traffic enters standby devices, asymmetric routing is supported. To implement the asymmetric routing functionality, configure both the redundant devices with a dedicated interface (interlink interface) for asymmetric traffic. This dedicated interface will redirect the traffic coming to the standby WAN interface to the active device.

The status of redundancy group members is determined through the use of hello messages sent over the control link. If either of the devices do not respond to a hello message within a configured time period, the software considers that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol. You can configure the following parameters for hello messages:

- Active timer.
- Standby timer.
- Hello time—The interval at which hello messages are sent.
- Hold time—The time period before which the active or standby device is declared to be down.

The hello time defaults to three seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine which pairs of interfaces are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is the RII that is associated with the interface.

Reasons for Switchover

Another factor that can cause a switchover is the priority setting that can be configured on each device. The device with the highest priority value will be the active device. If a fault occurs on either the active or the standby device, the priority of the device is decremented by a configurable amount, known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs and the standby device becomes the active device. You can override this default behavior by disabling the preemption attribute for the redundancy group. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of the redundancy group.

Each failure event that causes a modification of a redundancy group's priority generates a syslog entry that contains a time stamp, the redundancy group that was affected, the previous priority, the new priority, and a description of the failure event cause.

Another situation that can cause a switchover to occur is when the priority of a device or interface falls below the configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (this includes crashes).
- The run-time priority of the active device goes below that of the standby device.
- The run-time priority of the active device goes below the configured threshold level.
- The redundancy group on the active device is reloaded manually by using the **redundancy application reload group** *rg-number* command.
- Two consecutive hello messages missed on any monitored interface forces the interface into testing mode. Both devices will verify the link status on the interface and then execute the following tests:
  - Network activity test
  - Address Resolution Protocol (ARP) test
  - Broadcast ping test

Active/Active Failover

In an active/active failover configuration, both devices can process network traffic. Active/active failover generates virtual MAC (VMAC) addresses for interfaces in each redundancy group (RG).

One device in an active/active failover pair is designated as the primary (active) device, and the other is designated as the secondary (standby) device. Unlike with active/standby failover, this designation does not indicate which device becomes active when both devices start simultaneously. Instead, the primary/secondary designation determines the following:

- The device that provides the running configuration to the failover pair when they start simultaneously.
- The device on which the failover RG appears in the active state when devices start simultaneously. Each failover RG in the configuration is configured with a primary or secondary device preference. You can configure both failover RGs to be in the active state on a single device and the standby failover RGs to be on the other device. You can also configure one failover RG to be in the active state and the other RG to be in the standby state on a single device.

Active/Standby Failover

Active/standby failover enables you to use a standby device to take over the functionality of a failed device. A failed active device changes to the standby state, and the standby device changes to the active state. The device that is now in the active state takes over IP addresses and MAC addresses of the failed device and starts processing traffic. The device that is now in the standby state takes over standby IP addresses and MAC addresses. Because network devices do not see any change in the MAC-to-IP address pairing, Address Resolution Protocol (ARP) entries do not change or time out anywhere on the network.

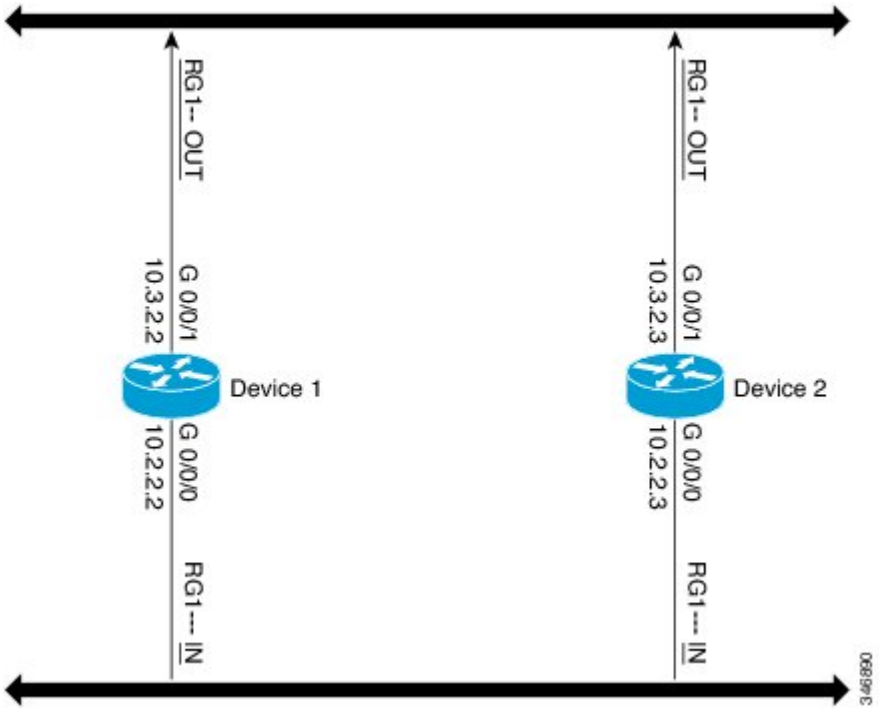
In an active/standby scenario, the main difference between two devices in a failover pair depends on which device is active and which device is a standby, namely which IP addresses to use and which device actively passes the traffic. The active device always becomes the active device if both devices start up at the same time (and are of equal operational health). MAC addresses of the active device are always paired with active IP addresses.

NAT Box-to-Box High-Availability LAN-LAN Topology

In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. The figure below shows the NAT box-to-box LAN-LAN topology. Network Address Translation (NAT) is in the active-standby mode and the peers are in one redundancy group (RG). All traffic or a subset of this traffic undergoes NAT translation.

Note Failover is caused by only those failures that the RG infrastructure listens to.

Figure 70: NAT Box-to-Box High-Availability LAN-LAN Topology



WAN-LAN Topology

In a WAN-LAN topology, two devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links.

WAN links can be provided by the same service provider or different service providers. In most cases, WAN links are provided by different service providers. To utilize WAN links to the maximum, configure an external device to provide a failover.

On LAN-based interfaces, a high availability virtual IP address is required to exchange client information and for faster failover. On WAN-based interfaces, the **redundancy group** *id* **ip** *virtual-ip* **decrement** *value* command is used for failover.

Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC. The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC. When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP and programs the interface's Ethernet controller to accept packets destined for the VMAC.

IPv6 Support

You can assign each redundancy group (RG) on a traffic interface for both IPv4 and IPv6 virtual IP (VIP) addresses under the same redundancy interface identifier (RII). Each RG uses a unique virtual MAC (VMAC) address per RII. For an RG, the IPv6 link-local VIP and global VIP coexist on an interface.

You can configure an IPv4 VIP, a link-local IPv6 VIP, and/or a global IPv6 VIP for each RG on a traffic interface. IPv6 link-local VIP is mainly used when configuring static or default routes, whereas IPv6 global VIP is widely used in both LAN and WAN topologies.

You must configure a physical IP address before configuring an IPv4 VIP.

FTP66 ALG Support Overview

Firewalls support the inspection of IPv6 packets and stateful Network Address Translation 64 (NAT64). For FTP to work over IPv6 packet inspection, the application-layer gateway (ALG) (also called the application-level gateway [ALG]), FTP66, is required. The FTP66 ALG is also called all-in-one FTP ALG and one FTP ALG.

The FTP66 ALG supports the following:

- Firewall IPv4 packet inspection
- Firewall IPv6 packet inspection
- NAT configuration
- NAT64 configuration (along with FTP64 support)
- NAT and firewall configuration
- NAT64 and firewall configuration

The FTP66 ALG has the following security vulnerabilities:

- Packet segmentation attack—The FTP ALG state machine can detect segmented packets, and the state machine processing is stopped until a complete packet is received.
- Bounce attack—The FTP ALG does not create doors (for NAT) or pinholes (for firewalls) with a data port number less than 1024. The prevention of a bounce attack is activated only when the firewall is enabled.

How to Configure Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Configuring a Redundancy Group Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**

4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*
7. **timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*}
8. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name prot1`	(Optional) Configures the redundancy group (RG) with a name.
Step 7	**timers hellotime** {*seconds*	**msec** *milliseconds*} **holdtime** {*seconds*	**msec** *milliseconds*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9`	Specifies the interval between when hello messages are sent and the time period before which a device is declared to be down.
Step 8	**authentication** {**text** *string*	**md5 key-string** [**0**	**7**] *key-string* **timeout** *seconds*	**key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies the authentication information.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]
9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.
Step 9	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:** `Device(config-red-app-grp)# track 200 decrement 200`	Specifies the priority value of a redundancy group that will be decremented if an event occurs.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Control Interface and a Data Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the redundancy group.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/2 protocol 1`	Specifies the control interface that is used by the redundancy group. • This interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time that a redundancy group will take to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a LAN Traffic Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **description** *string*
5. **encapsulation dot1q** *vlan-id*
6. **ip vrf forwarding** *name*
7. **ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **zone-member security** *zone-name*
9. **redundancy rii** *RII-identifier*
10. **redundancy group** *id* {**ip** *virtual-ip* | **ipv6** {*link-local-address* | *ipv6-address/prefix-length*} | **autoconfig**} [**exclusive**] [**decrement** *value*]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/0/2`	Configures an interface and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description lan interface`	(Optional) Adds a description to an interface configuration.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if)# encapsulation dot1q 18`	Sets the encapsulation method used by the interface.
Step 6	**ip vrf forwarding** *name* **Example:** `Device(config-if)# ip vrf forwarding trust`	Associates a VPN routing and forwarding (VRF) instance with an interface or subinterface. • The command will not be configured if the specified VRF is not configured.
Step 7	**ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

	Command or Action	Purpose
Step 8	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone member security z1`	Configures the interface as a zone member. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command while configuring a firewall. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 9	**redundancy rii** *RII-identifier* **Example:** `Device(config-if)# redundancy rii 100`	Configures an RII for redundancy group-protected traffic interfaces.
Step 10	**redundancy group** *id* {**ip** *virtual-ip* | **ipv6** {*link-local-address* | *ipv6-address/prefix-length*} | **autoconfig**} [**exclusive**] [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ipv6 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64 exclusive decrement 50`	Enables the redundancy group (RG) traffic interface configuration.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring a WAN Traffic Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *string*
5. **ipv6 address** {*ipv6-prefix/prefix-length* | *prefix-name sub-bits/prefix-length*}
6. **zone-member security** *zone-name*
7. **ip tcp adjust-mss** *max-segment-size*
8. **redundancy rii** *RII-identifier*
9. **redundancy asymmetric-routing enable**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description wan interface`	(Optional) Adds a description to an interface configuration.
Step 5	**ipv6 address** {*ipv6-prefix*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8:2222::/48`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z2`	Configures the interface as a zone member while configuring a firewall. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 7	**ip tcp adjust-mss** *max-segment-size* **Example:** `Device(config-if)# ip tcp adjust-mss 1360`	Adjusts the maximum segment size (MSS) value of TCP SYN packets going through a router.
Step 8	**redundancy rii** *RII-identifier* **Example:** `Device(config-if)# redundancy rii 360`	Configures an RII for redundancy group-protected traffic interfaces.

	Command or Action	Purpose
Step 9	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Associates a redundancy group with an interface that is used for asymmetric routing.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:**	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).

	Command or Action	Purpose
	`Device(config)# ip port-map ftp port 8090 list ipv6-acl`	
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:** `Device(config-cmap)# match access-group name ipv6-acl`	Configures the match criteria for a class map on the basis of the specified ACL.
Step 17	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Applying Zones to Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ipv6 address** *ipv6-address/prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security z2`	Creates a security zone and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*] **Example:** `Device(config)# zone-pair security in-2-out source z1 destination z2`	Creates a zone pair and enters security zone-pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy`	Attaches a policy map to a top-level policy map.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 11	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface.
Step 12	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 13	**zone-member security** *zone-name* **Example:** `Device(config-subif)# zone member security z1`	Configures the interface as a zone member. • For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of the

	Command or Action	Purpose
		zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface.
Step 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 15	**show policy-map type inspect zone-pair sessions** **Example:** `Device# show policy-map type inspect zone-pair sessions`	Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair. • The output of this command displays both IPv4 and IPv6 firewall sessions.

Example

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv4 address and vice versa:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:32847=>(209.165.201.2:21) ftp SIS_OPEN
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [37:84]

        Half-open Sessions
         Session 110D930C [2001:DB8:1::104]:32848=>(209.165.201.2:21) ftp SIS_OPENING
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv6 address:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:63=>[2001:DB8:2::102]:63 udp SIS_OPEN
```

```
Created 00:00:02, Last heard 00:00:01
Bytes sent (initiator:responder) [162:0]
```

Configuration Examples for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Example: Configuring a Redundancy Group Protocol

The following example shows how to configure a redundancy group with timers set for hello time and hold time messages:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover-threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 200 decrement 200
Device(config-red-app-grp)# end
```

Example: Configuring a Control Interface and a Data Interface

```
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# end
```

Example: Configuring a LAN Traffic Interface

```
Device# configure terminal
Device(config-if)# interface gigabitethernet 2/0/2
Device(config-if)# description lan interface
Device(config-if)# encapsulation dot1q 18
```

```
Device(config-if)# ip vrf forwarding trust
Device(config-if)# ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64
Device(config-if)# zone member security z1
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE exclusive
decrement 50
Device(config-if)# end
```

Example: Configuring a WAN Traffic Interface

The following example shows how to configure redundancy groups for a WAN-LAN scenario:

```
Device# configure terminal
Device(config-if)# interface gigabitethernet 2/1/0
Device(config-if)# description wan interface
Device(config-if)# ipv6 address 2001:DB8:2222::/48
Device(config-if)# zone-member security z2
Device(config-if)# ip tcp adjust-mss 1360
Device(config-if)# redundancy rii 360
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring Zones and Applying Zones to Interfaces

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
```

```
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
Device(config-if)# zone member security z1
Device(config-if)# end
```

Additional References for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 176: Feature Information for Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls

Feature Name	Releases	Feature Information
Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls	Cisco IOS XE Release 3.8S	The Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls feature supports high availability (HA) based on redundancy groups (RGs) on IPv6 firewalls. This feature enables you to configure pairs of devices to act as backup for each other. This feature can be configured to determine the active device based on a number of failover conditions. No commands were introduced or modified.
Box-to-Box High Availability Support for IPv6 Zone-Based Firewalls	Cisco IOS XE Release 3.8S	In Cisco IOS XE Release 3.10S, support was added for the Cisco ISR 4400 Series Routers.

CHAPTER 132

Firewall Stateful Inspection of ICMP

The Firewall Stateful Inspection of ICMP feature categorizes Internet Control Management Protocol Version 4 (ICMPv4) messages as either malicious or benign. The firewall uses stateful inspection to *trust* benign ICMPv4 messages that are generated within a private network and permits the entry of associated ICMP replies into the network. The Firewall Stateful Inspection of ICMP feature helps network administrators to debug network issues by using ICMP so that intruders cannot enter the network.

This module provides an overview of the firewall stateful inspection of ICMPv4 messages and describes how to configure the firewall to inspect ICMPv4 messages.

Prerequisites for Firewall Stateful Inspection of ICMP

- You must configure the Cisco firewall before you can configure the Firewall Stateful Inspection of ICMP feature.
- The network must allow all ICMP traffic to pass through security appliance interfaces.
- Access rules must be configured for ICMP traffic that terminates at a security appliance interface.

Restrictions for Firewall Stateful Inspection of ICMP

This feature does not work with the UDP traceroute utility, in which UDP datagrams are sent instead of ICMP packets. UDP traceroute is the default for UNIX systems. For a UNIX host to generate ICMP traceroute packets that are inspected by the firewall, use the "-I" option with the **traceroute** command.

Information About Firewall Stateful Inspection of ICMP

Overview of the Firewall Stateful Inspection of ICMP

Internet Control Management Protocol (ICMP) is a network protocol that provides information about a network and reports errors in the network. Network administrators use ICMP to debug network connectivity issues. To guard against potential intruders using ICMP to discover the topology of a private network, ICMPv4 messages can be blocked from entering a private network; however, network administrators may then be unable to debug the network.

You can configure Cisco routers to use access control lists (ACLs) to either completely allow or deny ICMPv4 messages. When using ACLs for ICMPv4 messages, message *inspection* has precedence over the configured allow or deny actions.

ICMPv4 messages that use the IP protocol can be categorized into the following two types:

- Informational messages that utilize a simple request/reply mechanism.
- Error messages that indicate that some sort of error has occurred while delivering an IP packet.

Note To prevent ICMP attacks from using the Destination Unreachable error message, only one Destination Unreachable message is allowed per session by the firewall.

A host that is processing a UDP session that is traversing the firewall may generate an ICMP error packet with a Destination Unreachable message. In such cases, only one Destination Unreachable message is allowed through the firewall for that session.

The following ICMPv4 packet types are supported:

Table 177: ICMPv4 Packet Types

Packet Type	Name	Description
0	Echo Reply	Reply to an echo request (type 8).
3	Unreachable	Possible reply to any request.
8	Echo Request	Ping or a traceroute request.
11	Time Exceeded	Reply if the time-to-live (TTL) size of a packet is zero.
13	Timestamp Request	Request.
14	Timestamp Reply	Reply to a timestamp request (type 13).

ICMPv4 packet types 0 and 8 are used to ping a destination; the source sends out an Echo Request packet and the destination responds with an Echo Reply packet. Packet types 0, 8, and 11 are used for ICMPv4 traceroute (that is, Echo Request packets that are sent start with a TTL size of 1) and the TTL size is incremented for

each hop. Intermediate hops respond to the Echo Request packet with a Time Exceeded packet and the final destination responds with an Echo Reply packet.

If an ICMPv4 error packet is an embedded packet, the embedded packet is processed according to the protocol and the policy configured for the packet. For example, if the embedded packet is a TCP packet, and a drop action is configured for the packet, the packet is dropped even if ICMPv4 has configured a pass action.

The following scenario describes how ICMPv4 packets pass through the firewall:

1. An ICMPv4 packet arrives at the source interface. The firewall uses the source and destination addresses of the packet without any change for packet inspection. The firewall uses IP addresses (source and destination), the ICMP type, and the protocol for session key creation and lookup.
2. The packet passes the firewall inspection.
3. Return traffic comes from the destination interface and, based on the ICMPv4 message type, the firewall creates the session lookup key.
4. **a.** If the reply message is an informational message, the firewall uses the source and destination addresses from the packet without any change for packet inspection. Here, the destination port is the ICMPv4 message request type.
   b. If the reply message is an ICMPv4 error message, the firewall uses the payload packet present in the ICMP error packet to create the session key for session lookup.
5. If the firewall session lookup is successful, the packet passes the firewall inspection.

ICMP Inspection Checking

ICMP return packets are checked by the inspect code, and not by access control lists (ACLs). The inspect code tracks destination address from each outgoing packet and checks each return packet. For Echo Reply and Timestamp Reply packets, the return address is checked. For Unreachable and Time Exceeded packets, the intended destination address is extracted from the packet data and checked.

How to Configure Firewall Stateful Inspection of ICMP

Configuring Firewall Stateful Inspection of ICMP

Perform this task to configure the firewall stateful inspection of ICMP, which includes the following:

- A class map that matches the ICMP traffic.
- A policy map with the inspect action.
- Security zones and zone pairs (to attach a firewall policy map to the zone pair).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard*

4. **class-map type inspect** *class-map-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class** *class-map-name*
9. **inspect**
10. **exit**
11. **exit**
12. **zone security** *zone-name*
13. **exit**
14. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
15. **service-policy type inspect** *policy-map-name*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* **Example:** `Device(config)# access-list 102 permit icmp 192.168.0.1 255.255.255.0 192.168.2.22 255.255.255.0`	Defines an extended IP access list.
Step 4	**class-map type inspect** *class-map-name* **Example:** `Device(config)# class-map type inspect c1`	Defines the class on which an action is to be performed and enters QoS class-map configuration mode.
Step 5	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol icmp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p1`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 8	**class** *class-map-name* **Example:** `Device(config-pmap)# class c1`	Defines the class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 9	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 11	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 12	**zone security** *zone-name* **Example:** `Device(config)# zone security z1`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 13	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 14	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security inout source z1 destination z2`	Creates a zone pair to which interfaces can be assigned and enters security zone-pair configuration mode.
Step 15	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p1`	Attaches a firewall policy map to a zone pair.

	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.

Verifying Firewall Stateful Inspection of ICMP

You can use the following **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ip access-lists**
3. **show policy-map type inspect** *policy-map-name*
4. **show policy-map type inspect zone-pair** *zone-pair-name*
5. **show zone security** *zone-name*
6. **show zone-pair security** [**source** *source-zone* **destination** *destination-zone*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip access-lists**

Example:

```
Device# show ip access-lists
```

Displays information about the specified policy map.

Step 3 **show policy-map type inspect** *policy-map-name*

Example:

```
Device# show policy-map type inspect p1
```

Displays information about the specified policy map.

Step 4 **show policy-map type inspect zone-pair** *zone-pair-name*

Example:

```
Device# show policy-map type inspect zone-pair inout
```

Displays the runtime inspect type policy-map statistics for the zone pair.

Step 5 **show zone security** *zone-name*

Example:

```
Device# show zone security z1
```

Displays zone security information.

Step 6 **show zone-pair security** [**source** *source-zone* **destination** *destination-zone*]

Example:

```
Device# show zone-pair security source z1 destination z2
```

Displays source and destination zones and the policy attached to the zone pair.

Example:

The following sample output from the **show ip access-lists** command shows how ACLs are created for an ICMP session for which only ping packets were issued from the host:

```
Device# show ip access-lists

 Extended IP access list 102
     permit icmp any host 192.168.133.3 time-exceeded
     permit icmp any host 192.168.133.3 unreachable
     permit icmp any host 192.168.133.3 timestamp-reply
     permit icmp any host 192.168.133.3 echo-reply (4 matches)
```

The following is sample output from the **show policy-map type inspect p1** command:

```
Device# show policy-map type inspect p1

 Policy Map type inspect p1
  Class c1
   Inspect
```

The following is sample output from the **show policy-map type inspect zone-pair inout** command:

```
Device# show policy-map type inspect zone-pair inout

 Zone-pair: inout
  Service-policy : p1
   Class-map: c1 (match-all)
    Match: protocol icmp
    Inspect
     Session creations since subsystem startup or last reset 0
     Current session counts (estab/half-open/terminating) [0:0:0]
     Maxever session counts (estab/half-open/terminating) [0:0:0]
     Last session created never
     Last statistic reset never
     Last session creation rate 0
     half-open session total 0
   Class-map: class-default (match-any)
    Match: any
    Drop
     0 packets, 0 bytes
```

The following is sample output from the **show zone security** command:

```
Device# show zone security
```

```
zone self
Description: System defined zone
```

The following is sample output from the **show zone-pair security** command:

```
Device# show zone-pair security source z1 destination z2

zone-pair name inout
  Source-Zone z1  Destination-Zone z2
  service-policy p1
```

Configuration Examples for Firewall Stateful Inspection of ICMP

Example: Configuring Firewall Stateful Inspection of ICMP

```
Device# configure terminal
Device(config)# access-list 102 permit icmp 192.168.0.1 255.255.255.0 192.168.2.22
255.255.255.0
Device(config)# class-map type inspect c1
Device(config-cmap)# match protocol icmp
Device(config-cmap)# exit
Device(config)# policy-map type inspect p1
Device(config-pmap)# class c1
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security inout source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect p1
Device(config-sec-zone-pair)# end
```

Additional References for Firewall Stateful Inspection of ICMP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards & RFCs

Standard/RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 950	*Internet Standard Subnetting Procedure*
RFC 1700	*Assigned Numbers*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Stateful Inspection of ICMP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 178: Feature Information for Firewall Stateful Inspection of ICMP

Feature Name	Releases	Feature Information
Firewall Stateful Inspection of ICMP	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2S	The Firewall Stateful Inspection of ICMP feature categorizes ICMPv4 messages as either malicious or benign. The firewall uses stateful inspection to *trust* benign ICMP messages that are generated within a private network and permits the entry of associated ICMP replies.

CHAPTER 133

LISP and Zone-Based Firewalls Integration and Interoperability

The LISP and Zone-Based Firewalls Integration and Interoperability feature enables inner-packet inspection of all Locator ID Separation Protocol (LISP) data packets that pass through a device. To enable LISP inner packet inspection, you have to configure the **lisp inner-packet inspection** command. Without LISP inner packet inspection, endpoint identifier (EID) devices in a LISP network will not have any firewall protection.

This module describes how to configure this feature.

Feature Information for LISP and Zone-Based Firewall Integration and Interoperability

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 179: Feature Information for LISP and Zone-Based Firewall Integration and Interoperability

Feature Name	Releases	Feature Information
LISP and Zone-Based Firewall Integration and Interoperability	Cisco IOS XE Release 3.13S	The LISP and Zone-Based Firewalls Integration and Interoperability feature enables inner-packet inspection of all Locator ID Separation Protocol (LISP) data packets that pass through a device. To enable LISP inner packet inspection, you have to configure the lisp inner-packet inspection command. Without LISP inner inspection, endpoint identifier (EID) devices in a LISP network will not have any firewall protection. The following commands were introduced or modified by this feature: **lisp inner-packet-inspection**, **show parameter-map type inspect-global**, and **show parameter-map type inspect global**.
Intrachassis and Interchassis High Availability for Zone-Based Firewall and LISP Integration	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, the LISP and Zone-Based Firewall Integration and Interoperability feature supports both intrachassis and interchassis high availability. No commands were introduced or modified by this feature.

Prerequisites for LISP and Zone-Based Firewall Integration and Interoperability

- The interchassis high availability configuration on active device and standby devices must be identical.

Restrictions for LISP and Zone-Based Firewall Integration and Interoperability

The following features are not supported:

- Locator ID Separator Protocol (LISP) mobility
- Zone-based firewall, LISP, and Web Cache Control Protocol (WCCP) interoperability
- Zone-based firewall and LISP subinterfaces with VRF interoperability

These features are not supported when LISP inner packet inspection is enabled:

- Asymmetric routing
- LISP control message inspection
- LISP inner packet fragmentation

- Network Address Translation (NAT) and NAT 64
- TCP reset
- Virtual routing and forwarding (VRF)
- Virtual TCP (vTCP)
- VRF-Aware Software Infrastructure (VASI)
- Web Cache Communication Protocol (WCCP)

Information About LISP and Zone-Based Firewalls Integration and Interoperability

LISP Overview

The Locator ID Separation Protocol (LISP) is a network architecture and protocol. LISP replaces a single IP address with two numbering spaces—Routing Locators (RLOCs), which are topologically assigned to network attachment points and used for routing and forwarding of packets through the network; and Endpoint Identifiers (EIDs), which are assigned independently from the network topology and used for numbering devices, and are aggregated along administrative boundaries.

LISP defines functions for mapping between the two numbering spaces and encapsulating traffic originated by devices using non-routable EIDs for transport across a network infrastructure that routes and forwards using RLOCs. LISP provides a set of functions for devices to exchange information that is used to map non-routable EIDs to routable RLOCs.

LISP requires LISP-specific configuration of one or more LISP-related devices, such as the LISP egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), map server (MS), and LISP alternative logical topology (ALT) device.

Zone-Based Firewall and LISP Interoperability Overview

The zone-based firewall can be deployed either on the southbound or northbound of the Locator ID Separator Protocol (LISP) xTR device, depending on where the edge router (routers such as Cisco ASR 1000 Aggregation Services Routers) is located in the network. The ingress tunnel router (ITR) and egress tunnel router (ETR) together are called the xTR device.

When the zone-based firewall is at the northbound of the xTR device; then the firewall can view LISP encapsulated packets, such as LISP tunneled packets, that pass through the network.

When the zone-based firewall is at the southbound of the xTR device, then the firewall can view the original packet. However; the firewall is not aware of any LISP xTR processing or do not see any LISP header. For egress packets, the xTR device does LISP encapsulation and adds the LISP header on top of the original packet after the firewall inspection. For ingress packets, the xTR device does LISP decapsulation (removal of the LISP header) before the firewall inspection and as a result, the firewall only inspects the original packet; and has no interaction with LISP at all.

This section describes the scenario when the zone-based firewall is deployed at the southbound of the LISP xTR device:

If an edge router is configured as a LISP xTR device to perform LISP encapsulation and decapsulation functions, you can configure the zone-based firewall between the LISP interface and the interfaces that face the LISP local endpoint identifier (EID) devices on the same edge router. LISP header decapsulation is performed before the header enters the zone-based firewall at the LISP interface. LISP header encapsulation is performed after the packet egresses from the firewall at the LISP interface. The firewall inspects only native traffic (what is native traffic here?) in the EID space.

This section describes the scenario when the zone-based firewall is deployed at the northbound of the LISP xTR devicce:

If more than one edge routers are deployed as load-sharing routers at the northbound of the xTR device, the firewall on the edge router is considered northbound of the xTR device. In this case, all packets that pass through the zone-based firewall are LISP encapsulated packets. When a packet arrives, the firewall inspects either the inner header or outer header of the LISP packets. By default, only the outer header is inspected. You can enable inner header inspection by using the **lisp inner-packet-inspection** command.

In Cisco IOS XE Release, if LISP inner packet inspection is enabled, the firewall only inspects the first fragmented inner packet, and all subsequent inner packets pass through the firewall without further inspection. If LISP inner packet inspection is enabled, the LISP instance ID is treated as virtual routing and forwarding (VRF) ID, and LISP packets that belong to different instance IDs are associated with different zone-based firewall sessions.

Feature Interoperability LISP

In Cisco IOS XE Release 3.13S, the LISP and Zone-Based Firewall Integration and Interoperability feature, works with the following features:

- IPv4 inner and outer headers
- IPv6 inner and outer headers
- LISP multitenancy
- Application layer gateways (ALGs)
- Application Inspection and Control (AIC)
- Mulitprotocol Label Switching (MPLS)
- In-Service Software Upgrade (ISSU)
- PxTR Case

Intrachassis and Interchassis High Availability for Zone-Based Firewall and LISP Integration

In Cisco IOS XE Release 3.14S, the LISP and Zone-Based Firewall Integration and Interoperability feature supports both intrachassis and interchassis high availability. When Location ID Separation Protocol (LISP) inner packet inspection is enabled, interchassis and intrachassis redundancy are supported at the xTR northbound device.

For LISP inner packet inspection at the northbound device, LISP instance ID is used as the virtual routing and forwarding (VRF) instance. The VRF configuration at northbound device is ignored if LISP inner packet inspection is enabled.

When two devices are located at the northbound of the xTR device and the xTR device is located inside the cloud, if LISP inner packet inspection is enabled on both devices, zone-based firewall sessions that are created for LISP inner packet flow is synced to the standby device.

A typical interchassis (box-to-box) high availability topology will have two devices in the routing locator (RLOC) space at the northbound of the xTR device. The xTR device sits in the inside network. If LISP inner packet inspection is enabled on both devices, zone-based firewall sessions that are created for LISP inner packets are synced to the standby device.

There are no configuration changes for intrachassis redundancy.

How to Configure LISP and Zone-Based Firewalls Integration and Interoperability

Enabling LISP Inner Packet Inspection

You can configure LISP inner packet inspection after configuring the **parameter-map type inspect global** command or the **parameter-map type inspect-global** command.

Note You cannot configure both these commands simultaneously.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **lisp inner-packet-inspection**
5. **end**
6. **show parameter-map type** {**inspect global** | **inspect-global**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect global** **Example:**	Configures a global inspect-type parameter map for connecting thresholds, timeouts, and other parameters

	Command or Action	Purpose
	`Device(config)# parameter-map type inspect global`	pertaining to the inspect action, and enters parameter-map type inspect configuration mode.
Step 4	**lisp inner-packet-inspection** **Example:** `Device(config-profile)# lisp inner-packet-inspection`	Enables LISP inner packet inspection.
Step 5	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.
Step 6	**show parameter-map type** {**inspect global** | **inspect-global**} **Example:** `Device# show parameter-map type inspect-global`	Displays global inspect-type parameter map information.

Example

The following sample output from the **show parameter-map type inspect-global** command displays that LISP inner-packet inspection is enabled:

```
Device# show parameter-map type inspect-global

parameter-map type inspect-global
  log dropped-packet off
  alert on
  aggressive aging disabled
  syn_flood_limit  unlimited
  tcp window scaling enforcement loose off
  max incomplete unlimited  aggressive aging disabled
  max_incomplete TCP unlimited
  max_incomplete UDP unlimited
  max_incomplete ICMP unlimited
  application-inspect all
  vrf default inspect vrf-default
  vrf vrf2 inspect vrf-default
  vrf vrf3 inspect vrf-default
  lisp inner-packet-inspection
```

Configuring Interchassis High Availability for LISP Inner Packet Inspection

Configuring the xTR Southbound Interface for Interchassis High Availability

Before you begin

Prerequisites

- Zones and zone-pairs must be configured.

- Redundancy and redundancy groups must be configured. See, the "Configuring Firewall Stateful Interchassis Redundancy" module in the *Zone-Based Policy Firewall Configuration Guide* for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **description** *string*
6. **ip address** *ip-address mask*
7. **exit**
8. **interface** *type number*
9. **description** *string*
10. **zone-member security** *zone-name*
11. **exit**
12. **interface** *type number*
13. **description** *string*
14. **ip address** *ip-address mask*
15. **zone-member security** *zone-name*
16. **cdp enable**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet 1/3/0`	Configures an interface and enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding lower`	Associates a VRF instance or a virtual network with an interface or subinterface.
Step 5	**description** *string* **Example:**	Adds a description to an interface configuration. • The zone-based firewall cannot be configured at this interface.

	Command or Action	Purpose
	`Device(config-if)# description facing RLOC and the LISP cloud; has a LISP header.`	
Step 6	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.1.27 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface LISP 0`	Configures an interface and enters interface configuration mode. • This is the LISP virtual interface.

Step 9	**description** *string*
	Example:
	`Device(config-if)# description LISP virtual interface.`
	`Adds LISP header after firewall inspection or removes LISP header before firewall inspection.`
		Adds a description to an interface configuration.

Step 10	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-3a`	Attaches an interface to a security zone.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**interface** *type number* **Example:** `Device(config)# interface tengigabitethernet 0/3/0`	Configures an interface and enters interface configuration mode.
Step 13	**description** *string* **Example:** `Device(config-if)# description facing internal network, does not have a LISP header.`	Adds a description to an interface configuration.
Step 14	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 15	**zone-member security** *zone-name* **Example:**	Attaches an interface to a security zone.

	Command or Action	Purpose
	`Device(config-if)# zone-member security ge0-0-0`	
Step 16	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enable Cisco Discovery Protocol (CDP) on an interface.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the xTR Northbound Interface for LISP Inner Packet Inspection

In this configuration, a Locator ID Separation Protocol (LISP) virtual interface is not needed because at northbound the LISP header is not inspected. However, you can configure the zone-based firewall to inspect either LISP inner packets or outer packets.

Before you begin

- Zones and zone-pairs must be configured.
- Redundancy and redundancy groups must be configured. See, the "Configuring Firewall Stateful Interchassis Redundancy" module in the *Zone-Based Policy Firewall Configuration Guide* for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **description** *string*
5. **ip address** *ip-address mask*
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **redundancy rii** *id*
9. **redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value*
10. **exit**
11. **interface** *type number*
12. **description** *string*
13. **ip address** *ip-address mask*
14. **zone-member security** *zone-name*
15. **negotiation auto**
16. **redundancy rii** *id*
17. **redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value*
18. **ip virtual-reassembly**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/2/1`	Configures an interface and enters interface configuration mode. • This interface can see the entire LISP packet.
Step 4	**description** *string* **Example:** `Device(config-if)# description RLOC-space/north LAN`	Adds a description to an interface configuration.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.8 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-3`	Attaches an interface to a security zone.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 8	**redundancy rii** *id* **Example:** `Device(config-subif)# redundancy rii 200`	Configures the redundancy interface identifier (RII) for redundancy group protected traffic interfaces
Step 9	**redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value* **Example:** `Device(config-if)# redundancy group 1 ip 198.51.100.12 exclusive decrement 50`	Enables the redundancy group (RG) traffic interface configuration.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/3`	Configures an interface and enters interface configuration mode. • This interface can see the entire LISP packet.
Step 12	**description** *string* **Example:** `Device(config-if)# description RLOC-space/south LAN`	Adds a description to an interface configuration.
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.27 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security ge0-0-0`	Attaches an interface to a security zone.
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 16	**redundancy rii** *id* **Example:** `Device(config-subif)# redundancy rii 300`	Configures the redundancy interface identifier (RII) for redundancy group protected traffic interfaces
Step 17	**redundancy group** *id* **ip** *virtual-ip* **exclusive decrement** *value* **Example:** `Device(config-if)# redundancy group 1 ip 194.88.4.1 exclusive decrement 50`	Enables the RG traffic interface configuration.
Step 18	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables virtual fragment reassembly (VFR) on an interface.
Step 19	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for LISP and Zone-Based Firewalls Integration and Interoperability

Example: Enbaling LISP Inner Packet Inspection

```
Device# configure terminal
Device(config)# parameter-map type inspect-global
Device(config-profile)# lisp inner-packet-inspection
Device(config-profile)# end
```

The following example shows a zone-based firewall configuration with LISP inner-packet inspection enabled:

```
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family

class-map type inspect match-any c-ftp-tcp
 match protocol ftp
 match protocol telnet
 match protocol http
 match protocol tcp
 match protocol udp
!
policy-map type inspect p1
 class type inspect c-ftp-tcp
  inspect
 class class-default
!
zone security ge0-0-0
!
zone security ge0-0-3
!
zone-pair security zp-ge000-ge003 source ge0-0-0 destination ge0-0-3
 service-policy type inspect p1
!
zone-pair security zp-ge003-ge000 source ge0-0-3 destination ge0-0-0
 service-policy type inspect p1
!
interface TenGigabitEthernet 1/3/0
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:100::2/64
 zone-member security ge0-0-0
!
interface TenGigabitEthernet 0/3/0
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:200::2/64
 zone-member security ge0-0-3
!
parameter-map type inspect global
 lisp inner-packet-inspection
 log dropped-packet off
  alert on
```

!

Configuring Interchassis High Availability for LISP Inner Packet Inspection

Additional References for LISP and Zone-Based Firewalls Integration and Interoperability

Related Documents

Related Topic	Document Title
Cisco commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
LISP commands	Cisco IOS IP Routing: LISP Command Reference
LISP configuration guide	*IP Routing: LISP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

CHAPTER 134

Application Aware Firewall

This document describes how Zone Based FireWall policy is defined based on the applications that NBAR can detect and make Zone Based FireWall application aware. The Application FireWall inspects the traffic and blocks traffic based on applications, category, application-family or application-group. This application aware firewall feature provides the following benefits:

- Application visibility and granular control
- Classification of 1400+ layer 7 applications
- Allows or blocks traffic by application, category, application-family or application-group

Feature Information for Application Aware Firewall

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
Application Aware Zone-based FW	Cisco IOS XE Fuji 16.9.1	This document describes how Zone Based FireWall policy is defined based on the applications that NBAR can detect and make Zone Based FireWall application aware. The Application FireWall inspects the traffic and blocks traffic based on applications, category, application-family or application-group. The following commands were introduced or modified: **show class-map***avc-classmap-name* **show policy-map type inspect zone-pair** **show policy-map type inspect zone-pair sessions** **show policy-map type inspect avc** **show platform hardware qfpactive feature firewall drop**

Information About Application Awareness on Zone-Based FW

Prerequisites for Application Aware Firewall

- Ensure that traffic is matched to the Layer3/Layer4 inspect class map. If the traffic does not match the firewall inspection, the AVC policy fails to see the traffic.
- Inspect DNS in the same class-map where the AVC service-policy is applied.

Restrictions on Application Aware Zone-Based FW

- No support for traffic to self-zone.
- The AVC inspect policy should allow all and only deny certain application because many applications are interdependent and therefore allowing one application while denying all others do not work all the time.
- Each application class-map can have upto 16 filters (each match is considered a filter).
- The AVC policy-map can have upto 32 class-maps (including class-default).
- You cannot configure **match protocol attribute application-family** or **match protocol attribute application-group** if you specify the category using the **match protocol attribute category** command.

Before you configure class-map and policy-map, use the **parameter-map type inspect** configure the parameter-map type to log dropped packets:

```
Device (config)# parameter-map type inspect
Device (config-map)# log dropped-packets
```

Policies Based on Network Layers L3/L4

Zone-based Firewall uses policies based on network layers L3/L4, for example, class maps are based on ACL and L4 protocolsTCP/UDP/ICMP or L7 protocolsFTP and SIP. Policies that are defined using the L7 protocol utilize the protocol's destination port to classify the packet. ZBF lacks application visibility, it supports FTP inspection through the FTP ALG, and only identifies the protocols that are based on port 21.

Note If an FTP control flow is opened on some random port, zone-based firewall cannot identify the application.

How to Configure NBAR Based Application Awareness on ZBFW

Configure Layer 4 Zone-Based Firewall

```
Device(config-profile)#class-map type inspect match-any cm1
Device(config-cmap)#match protocol http
Device(config-cmap)#match protocol https
Device(config-cmap)#match protocol dns
Device(config-cmap)#match protocol tcp
Device(config-cmap)#match protocol udp
Device(config-cmap)#match protocol icmp
Device(config-cmap)#exit
Device(config)#class-map match-any nbar-class1
Device(config-cmap)#match protocol yahoo-mail
Device(config-cmap)#match protocol amazon
Device(config-cmap)#match protocol attribute category consumer-internet
Device(config-cmap)#exit
```

L7 Service Policy for Application Aware Firewall

Procedure

	Command or Action	Purpose

Step 1	Configure the class-map for inspection.
	Example:
	`class-map type inspect match-any cm1`
	` match protocol http`
	` match protocol https`
	` match protocol dns`
	` match protocol tcp`
	` match protocol udp`
	` match protocol icmp`
		Defines the protocols and category using the **class-map type inspect** and **match protocol** commands.

Step 2	Define the action, in this case the AVC, using the application firewall policy.
	Example:
	`policy-map type inspect avc nbar-policy1`
	`class nbar-class1`
		Uses the **deny** command to refuse the remote network management protocols listed in the `nbar-class1`class map.

	Command or Action	Purpose
	deny class class-default allow	

Step 3	Log the dropped packets using the application firewall policy.
	Example:
	policy-map type inspect pm1
	class type inspect cm1
	  inspect
	  service-policy avc nbar-policy1
	class class-default
	  drop log
	Traffic from amazon, in nbar-class1, is denied by the policy. For example, a dropped packet is shown in the following drop log message:
	Oct 17 12:44:08.101: %IOSXE-6-PLATFORM: R0/0:
	cpp_cp:
	QFP:0.0 Thread:000 TS:00000002517650404876
	%FW-6-DROP_PKT:
	Dropping dns/amazon pkt from GigabitEthernet3
	171.70.168.183:53 =>
	171.10.1.101:50877(target:class)
	-(in_to_out:cm1) due to AVC Policy drop:classify
	result with ip ident 65434	

What to do next

Add the **ip nbar protocol-discovery ipv4** command on the ingress interface. Then use the **show ip nbar protocol-discovery interface** [**intf-name**] command to see the application classification.

Example: Application Aware Show Commands

In this example, the **show policy-map type inspect zone-pair** command shows the policy map statistics and other information including information about the sessions existing on a specified zone pair. The line following `Class-map: nbar-class1 (match-any)` includes the packet counter value (`7 packets`), which increases whenever traffic matches the nbar-class1 class.

```
Device# show policy-map type inspect zone-pair

Zone-pair: in_to_out
Service-policy inspect : pm1

Class-map: cm1 (match-any)
Match: protocol http
Match: protocol https
Match: protocol dns
Match: protocol tcp
Match: protocol udp
Match: protocol icmp
Inspect
Packet inspection statistics [process switch:fast switch]
tcp packets: [0:485]
dns packets: [0:51]
```

```
Session creations since subsystem startup or last reset 21
Current session counts (estab/half-open/terminating) [13:0:0]
Maxever session counts (estab/half-open/terminating) [13:2:0]
Last session created 00:00:00
Last statistic reset 00:00:19
Last session creation rate 151
Last half-open session total 0

Service-policy inspect avc : nbar-policy1

Class-map: nbar-class1 (match-any)
7 packets, 1449 bytes
30 second offered rate 1000 bps, drop rate 0000 bps
Match: protocol amazon
Match: protocol yahoo-mail
Match: protocol attribute category consumer-internet
Deny

Class-map: class-default (match-any)
211 packets, 94091 bytes
30 second offered rate 27000 bps, drop rate 0000 bps
Match: any
Allow

Class-map: class-default (match-any)
Match: any
Drop
0 packets, 0 bytes
```

```
Device# show platform hardware qfp active feature firewall drop
-------------------------------------------------------------------------------
Drop Reason                                                            Packets
-------------------------------------------------------------------------------
AVC Policy drop:classify result                                             38
```

```
Device# show platform hardware qfp active feature firewal datapath scb
[s=session  i=imprecise channel c=control channel  d=data channel A/D=appfw action allow/deny]
Session ID:0x0000DA5B  171.10.1.101 64204 171.70.168.183 53 proto 17 (0:0) (1456:0xd000208)
 [scA]
Session ID:0x0000DA18  171.10.1.101 58836 74.125.199.103 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA5A  171.10.1.101 64206 8.8.8.8 53 proto 17 (0:0) (0:0xd000001) [sc]
Session ID:0x0000DA11  171.10.1.101 58833 74.125.199.84 443 proto 6 (0:0) (1440:0xd000210)
 [sdA]
Session ID:0x0000DA57  171.10.1.101 64205 173.36.131.10 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
Session ID:0x0000DA2C  171.10.1.101 58839 74.125.199.94 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA59  171.10.1.101 64203 173.36.131.10 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
Session ID:0x0000DA0B  171.10.1.101 58831 74.125.199.94 443 proto 6 (0:0) (1456:0xd000208)
 [sdA]
Session ID:0x0000DA5C  171.10.1.101 64207 8.8.4.4 53 proto 17 (0:0) (0:0xd000001) [sc]
Session ID:0x0000DA58  171.10.1.101 64203 171.70.168.183 53 proto 17 (0:0) (1761:0xd00033f)
 [scD]
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 135

Firewall Support of Skinny Client Control Protocol

The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and the Skinny Client Control Protocol (SCCP). Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This feature extends the support of SCCP to accommodate video channels.

Prerequisites for Firewall Support of Skinny Client Control Protocol

- Your system must be running Cisco IOS XE Release 2.1 or a later release.
- You must enable the firewall for the SCCP application-level gateway (ALG) to work.
- You must enable the TFTP ALG for SCCP to work because IP phones that use Skinny need the TFTP configuration file from the Cisco Unified Communications Manager.

Restrictions for Firewall Support of Skinny Client Control Protocol

- IPv6 address inspection and translation is not supported.
- TCP segmentation is not supported.

Information About Firewall Support of Skinny Client Control Protocol

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SCCP Inspection Overview

SCCP inspection enables voice communication between two SCCP clients by using the Cisco Unified Communications Manager. The Cisco Unified Communications Manager uses the TCP port 2000 (the default SCCP port) to provide services to SCCP clients. Initially, the SCCP client connects to the primary Cisco Unified Communications Manager by establishing a TCP connection and, if available, connects to a secondary Cisco Unified Communications Manager. After the TCP connection is established, the SCCP client registers with the primary Cisco Unified Communications Manager, which is used as the controlling Cisco Unified Communications Manager until it reboots or a keepalive failure occurs. Thus, the TCP connection between the SCCP client and the Cisco Unified Communications Manager exists forever and is used to establish calls coming to or from the client. If a TCP connection fails, the secondary Cisco Unified Communications Manager is used. All data channels established with the initial Cisco Unified Communications Manager remain active and will be closed after the call ends.

The SCCP protocol inspects the locally generated or terminated SCCP control channels and opens or closes pinholes for media channels that originate from or are destined to the firewall. Pinholes are ports that are opened through a firewall to allow an application controlled access to a protected network.

The table below lists the set of messages that are necessary for the data sessions to open and close. SCCP inspection will examine the data sessions that are used for opening and closing the access list pinholes.

Table 180: SCCP Data Session Messages

Skinny Inspection Message	Description
CloseReceiveChannel	Indicates that the call should be aborted. Any intermediate sessions created by the firewall and NAT have to be cleaned up when this message is received.
OpenReceiveChannelACK	Indicates that the phone is acknowledging the OpenReceiveChannel message that it received from the Cisco Unified Communications Manager.
StartMediaTransmission	Contains the Realtime Transport Protocol (RTP) information of the phone that is the source or destination of the call. The message contains the IP address, the RTP port that the other phone is listening on, and the Call ID that uniquely identifies the call.
StopMediaTransmission	Indicates that the call has ended. Sessions can be cleaned up after receiving this message.
StationCloseReceiveChannel	Instructs the Skinny client (on the basis of the information in this message) to close the receiving channel.
StationOpenMultiMediaReceiveChannelAck	Contains the IP address and port information of the Skinny client sending this message. It also contains the status of whether the client is willing to receive video and data channels.
StationOpenReceiveChannelAck	Contains the IP address and port information of the Skinny client sending this message. This message also contains the status of whether or not the client is willing to receive voice traffic.
StationStartMediaTransmission	Contains the IP address and port information of the remote Skinny client.
StationStartMultiMediaTransmit	Indicates that the Cisco Unified Communications Manager received an OpenLogicalChannelAck message for the video or the data channel.
StationStopMediaTransmission	Instructs the Skinny client (on the basis of the information in this message) to stop transmitting voice traffic.
StationStopSessionTransmission	Instructs the Skinny client (on the basis of the information in this message) to end the specified session.

ALG--SCCP Version 17 Support

The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and the IP phones that use Cisco Unified Communications

Manager 7.0 support only SCCP Version 17 messages. The format of SCCP changed from Version 17 to support IPv6. The SCCP ALG checks for the SCCP version in the prefix of a message before parsing it according to the version. The SCCP message version is extracted from the message header and if it is greater than Version 17, the message is parsed by using the Version 17 format and the IPv4 address and port information is extracted. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.

Note IPv6 address inspection and translation are not supported.

The IP address format of the following SCCP ALG-handled messages changed in Version 17:

- StationOpenMultiMediaReceiveChannelAck
- StationOpenReceiveChannelAckMessage
- StationRegisterMessage
- StationStartMediaTransmissionAckMessage
- StationStartMultiMediaTransmissionAckMessage
- StationStartMediaTransmissionMessage
- StationStartMultiMediaTransmissionMessage

How to Configure Firewall Support of Skinny Client Control Protocol

Configuring a Skinny Class Map and Policy Map

When you enable SCCP (through the **match protocol** command) in a firewall configuration, you must enable TFTP (through the **match protocol** command); otherwise, the IP phones that use SCCP cannot communicate with the Cisco Unified Communications Manager. SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **exit**

11. **class class-default**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any cmap1`	Creates an inspect type class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol skinny`	Configures the match criterion for a Skinny class map.
Step 5	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol tftp`	Configures the match criterion for a TFTP class map.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pmap1`	Creates an inspect type policy map and enters policy map configuration mode.
Step 8	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect cmap1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 9	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode and enters policy map configuration mode.

	Command or Action	Purpose
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 12	**end** **Example:** `Router(config-pmap)# end`	Exits policy map configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an SCCP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Router(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect pmap1`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall Support of Skinny Control Protocol

Example: Configuring an SCCP Class Map and a Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any cmap1
Router(config-cmap)# match protocol skinny
Router(config-cmap)# match protocol tftp
Router(config-cmap)# exit
Router(config)# policy-map type inspect pmap1
Router(config-pmap)# class type inspect cmap1
Router(config-pmap-c)# inspect
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# end
```

Example: Configuring a Zone Pair and Attaching an SCCP Policy Map

```
Router# configure terminal
Router(config)# zone security zone1
Router(config-sec-zone)# exit
Router(config)# zone security zone2
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source zone1 destination zone2
Router(config-sec-zone-pair)# service-policy type inspect pmap1
Router(config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/0/0
```

```
Router(config-if)# zone-member security zone1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/1/1
Router(config-if)# zone-member security zone2
Router(config-if)# end
```

Additional References for Firewall Support of Skinny Client Control Protocol

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Support for Skinny Client Control Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 181: Feature Information for Firewall Support for Skinny Client Control Protocol

Feature Name	Releases	Feature Information
ALG—SCCP V17 Support	Cisco IOS XE Release 3.5S	The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP version 17 packets. The SCCP format has changed from version 17 to support IPv6.
Firewall—SCCP Video ALG Support	Cisco IOS XE Release 2.4	SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager. This feature enables Cisco firewalls to inspect Skinny control packets that are exchanged between a Skinny client and the Cisco Unified Communications Manager. The following command was modified: **match protocol**.

Feature Name	Releases	Feature Information
Firewall Support for Skinny Client Control Protocol	Cisco IOS XE Release 2.1	The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and SCCP. Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This feature extends the support of SCCP to accommodate video channels..

CHAPTER 136

IPv6 Zone-Based Firewall Support over VASI Interfaces

This feature supports VRF-Aware Service Infrastructure (VASI) interfaces over IPv6 firewalls. This feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to traffic that flows across two different virtual routing and forwarding (VRF) instances. VASI interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs). VASI interfaces support IPv4 and IPv6 unicast traffic.

This module provides information about VASI interfaces and describes how to configure VASI interfaces.

Restrictions for IPv6 Zone-Based Firewall Support over VASI Interfaces

- Multiprotocol Label Switching (MPLS) traffic over VRF-Aware Software Infrastructure (VASI) interfaces is not supported.
- IPv4 and IPv6 multicast traffic is not supported.
- VASI interfaces do not support the attachment of queue-based features. The following commands are not supported on modular QoS CLI (MQC) policies that are attached to VASI interfaces:
  - **bandwidth (policy-map class)**
  - **fair-queue**
  - **priority**
  - **queue-limit**
  - **random-detect**
  - **shape**

Information About IPv6 Zone-Based Firewall Support over VASI Interfaces

VASI Overview

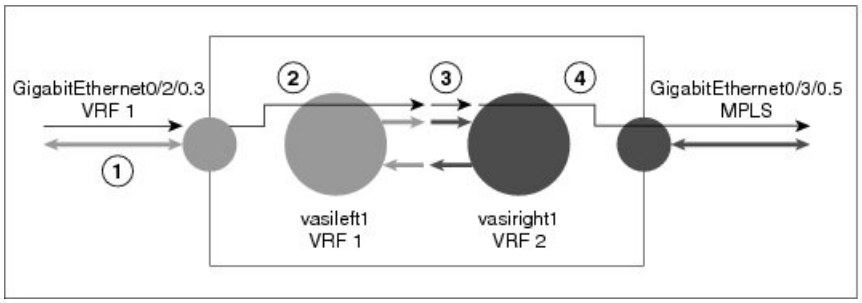
VRF-Aware Software Infrastructure (VASI) provides the ability to apply services such as, a firewall, GETVPN, IPsec, and Network Address Translation (NAT), to traffic that flows across different virtual routing and forwarding (VRF) instances. VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF instance. The VASI virtual interface is the next-hop interface for any packet that needs to be switched between these two VRF instances. VASI interfaces provide the framework to configure a firewall or NAT between VRF instances.

Each interface pair is associated with two different VRF instances. The pairing is done automatically based on the two interface indexes such that the vasileft interface is automatically paired to the vasiright interface. For example, in the figure below, vasileft1 and vasiright1 are automatically paired, and a packet entering vasileft1 is internally handed over to vasiright1.

On VASI interfaces, you can configure either static routing or dynamic routing with Internal Border Gateway Protocol (IBGP), Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF).

The following figure shows an inter-VRF VASI configuration on the same device.

Figure 71: Inter-VRF VASI Configuration



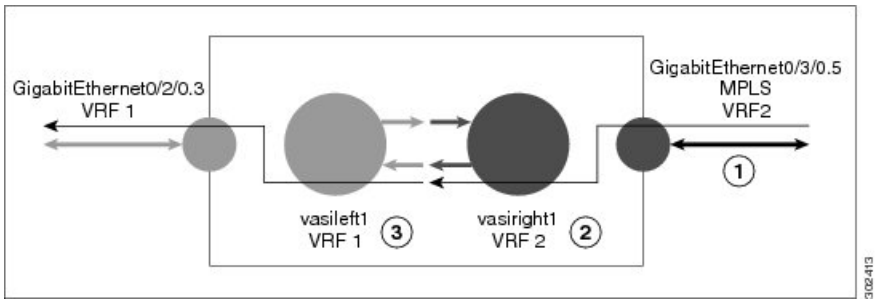
When an inter-VRF VASI is configured on the same device, the packet flow happens in the following order:

1. A packet enters the physical interface that belongs to VRF 1 (Gigabit Ethernet 0/2/0.3).
2. Before forwarding the packet, a forwarding lookup is done in the VRF 1 routing table. Vasileft1 is chosen as the next hop, and the Time to Live (TTL) value is decremented from the packet. Usually, the forwarding address is selected on the basis of the default route in the VRF. However, the forwarding address can also be a static route or a learned route. The packet is sent to the egress path of vasileft1 and then automatically sent to the vasiright1 ingress path.
3. When the packet enters vasiright1, a forwarding lookup is done in the VRF 2 routing table, and the TTL is decremented again (second time for this packet).
4. VRF 2 forwards the packet to the physical interface, Gigabit Ethernet 0/3/0.5.

The following figure shows how VASI works in a Multiprotocol Label Switching (MPLS) VPN configuration.

Note In the following figure, MPLS is enabled on the Gigabit Ethernet interface, but MPLS traffic is not supported across VASI pairs.

Figure 72: VASI with an MPLS VPN Configuration



When VASI is configured with a Multiprotocol Label Switching (MPLS) VPN, the packet flow happens in the following order:

1. A packet arrives on the MPLS interface with a VPN label.
2. The VPN label is stripped from the packet, a forwarding lookup is done within VRF 2, and the packet is forwarded to vasiright1. The TTL value is decremented from the packet.
3. The packet enters vasileft1 on the ingress path, and another forwarding lookup is done in VRF 1. The packet is sent to the egress physical interface in VRF1 (Gigabit Ethernet 0/2/0.3). The TTL is again decremented from the packet.

How to Configure IPv6 Zone-Based Firewall Support over VASI Interfaces

Configuring VRFs and Address Family Sessions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and enters VRF configuration mode.
Step 6	**end** **Example:** `Device(config-vrf)# end`	Exits VRF configuration mode and enters privileged EXEC mode.

Configuring Class Maps and Policy Maps for VASI Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **class-map type inspect match-any** *class-map-name*
5. **match protocol** *name*
6. **match protocol** *name*
7. **exit**
8. **policy-map type inspect** *policy-map-name*
9. **class type inspect** *class-map-name*
10. **inspect**
11. **exit**
12. **class class-default**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6-unicast routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any c-map`	Creates an inspect type class map and enters QoS class-map configuration mode.
Step 5	**match protocol** *name* **Example:** `Device(config-cmap)# match protocol icmp`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 6	**match protocol** *name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of a specified protocol.
Step 7	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 8	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect p-map`	Creates a protocol-specific inspect-type policy map and enters QoS policy-map configuration mode.
Step 9	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect c-map`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 10	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 11	**exit** **Example:**	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c)# exit`	
Step 12	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class and enters QoS policy-map class configuration mode. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring Zones and Zone Pairs for VASI Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
6. **service-policy type inspect** *policy-map-name*
7. **exit**
8. **interface** *type number*
9. **vrf forwarding** *vrf-name*
10. **no ip address**
11. **zone member security** *zone-name*
12. **ipv6 address** *ipv6-address*/*prefix-length*
13. **ipv6 enable**
14. **negotiation auto**
15. **exit**
16. **interface** *type number*
17. **no ip address**
18. **ipv6 address** *ipv6-address*/*prefix-length*
19. **ipv6 enable**
20. **negotiation auto**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security in`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security in-out source in destination out`	Creates a zone pair and enters security zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 6	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect p-map`	Attaches a policy map to a top-level policy map. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 7	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 9	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VRF1`	Associates a virtual routing and forwarding (VRF) instance or a virtual network with an interface or subinterface.
Step 10	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.

	Command or Action	Purpose
Step 11	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security in`	Attaches an interface to a security zone.
Step 12	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 13	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 14	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 15	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 17	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 18	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:3:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 19	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 20	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables advertisement of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 21	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring VASI Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ipv6-address/prefix-length* **link-local**
6. **ipv6 address** *ipv6-address/prefix-length*
7. **ipv6 enable**
8. **no keepalive**
9. **zone member security** *zone-name*
10. **exit**
11. **interface** *type number*
12. **ipv6 address** *ipv6-address/prefix-length* **link-local**
13. **ipv6 address** *ipv6-address/prefix-length*
14. **ipv6 enable**
15. **no keepalive**
16. **exit**
17. **ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number ipv6-address*
18. **ipv6 route vrf** *vrf-name ipv6-prefix/prefix-length interface-type interface-number ipv6-address*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vasileft 1`	Configures a VASI interface and enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VRF1`	Associates a virtual routing and forwarding (VRF) instance or a virtual network with an interface or subinterface.
Step 5	**ipv6 address** *ipv6-address/prefix-length* **link-local** **Example:**	Configures an IPv6 link-local address for an interface and enable IPv6 processing on the interface.

	Command or Action	Purpose
	`Device(config-if)# ipv6 address FE80::8EB6:4FFF:FE6C:E701 link-local`	
Step 6	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:4:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 8	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables keepalive packets.
Step 9	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security out`	Attaches an interface to a security zone.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface vasiright 1`	Configures a VASI interface and enters interface configuration mode.
Step 12	**ipv6 address** *ipv6-address/prefix-length* **link-local** **Example:** `Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local`	Configures an IPv6 link-local address for an interface and enable IPv6 processing on the interface.
Step 13	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:4:1234/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 14	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 15	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables keepalive packets.
Step 16	**exit** **Example:**	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 17	**ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number ipv6-address* **Example:** `Device(config)# ipv6 route 2001::/64 vasileft 1 2001::/64`	Establishes static IPv6 routes.
Step 18	**ipv6 route vrf** *vrf-name ipv6-prefix/prefix-length interface-type interface-number ipv6-address* **Example:** `Device(config)# ipv6 route vrf vrf1 2001::/64 vasiright 1 2001::/64`	Specifies all VRF tables or a specific VRF table for an IPv6 address.
Step 19	**end** **Example:** `Device(config# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for IPv6 Zone-Based Firewall Support over VASI Interfaces

Example: Configuring VRFs and Address Family Sessions

```
Device# configure terminal
Device(config)# vrf definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# end
```

Example: Configuring Class Maps and Policy Maps for VASI Support

```
Device# configure terminal
Device(config)# ipv6-unicast routing
Device(config)# class-map type inspect match-any c-map
Device(config-cmap)# match protocol icmp
Device(config-cmap)# match protocol tcp
Device(config-cmap)# match protocol udp
Device(config-cmap)# exit
Device(config)# policy-map type inspect p-map
Device(config-pmap)# class type inspect c-map
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# end
```

Example: Configuring Zones and Zone Pairs for VASI Support

```
Device# configure terminal
Device(config)# zone security in
Device(config)# exit
Device(config)# zone security out
Device(config)# exit
Device(config)# zone-pair security in-out source in destination out
Device(config-sec-zone-pair)# service-policy type inspect p-map
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# vrf forwarding VRF1
Device(config-if)# no ip address
Device(config-if)# zone member security in
Device(config-if)# ipv6 address 2001:DB8:2:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# ipv6 address 2001:DB8:3:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# negotiation auto
Device(config-if)# end
```

Example: Configuring VASI Interfaces

```
Device# configure terminal
Device(config)# interface vasileft 1
Device(config-if)# vrf forwarding VRF1
Device(config-if)# ipv6 address FE80::8EB6:4FFF:FE6C:E701 link-local
Device(config-if)# ipv6 address 2001:DB8:4:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# no keepalive
Device(config-if)# zone-member security out
Device(config-if)# exit
Device(config)# interface vasiright 1
Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local
Device(config-if)# ipv6 address 2001:DB8:4:1234/64
Device(config-if)# ipv6 enable
Device(config-if)# no keepalive
Device(config-if)# exit
Device(config)# ipv6 route 2001::/64 vasileft 1 2001::/64
Device(config)# ipv6 route vrf vrf1 2001::/64 vasiright 1 2001::/64
Device(config)# end
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Zone-Based Firewall Support over VASI Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 182: Feature Information for IPv6 Zone-Based Firewall Support VASI Interfaces

Feature Name	Releases	Feature Information
IPv6 Zone-Based Firewall Support over VASI Interfaces	Cisco IOS XE Release 3.7S	This feature supports VASI interfaces over IPv6 firewalls. This feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to traffic that flows across two different virtual routing and forwarding (VRF) instances. VASI interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs). VASI interfaces support IPv4 and IPv6 unicast traffic. No commands were introduced or modified for this feature.

CHAPTER 137

Configuring the VRF-Aware Software Infrastructure

The VRF-Aware Software Infrastructure feature allows you to apply services such as, access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls, to traffic that flows across two different virtual routing and forwarding (VRF) instances. VRF-Aware Software Infrastructure (VASI) interfaces support the redundancy of Route Processors (RPs) and Forwarding Processors (FPs), IPsec, and IPv4 and IPv6 unicast and multicast traffic.

This module describes how to configure VASI interfaces.

Restrictions for Configuring the VRF-Aware Software Infrastructure

- Multiprotocol Label Switching (MPLS) traffic over VRF-Aware Software Infrastructure (VASI) interfaces is not supported.
- VASI interfaces do not support the attachment of queue-based features. The following commands are not supported on Modular QoS CLI (MQC) policies that are attached to VASI interfaces:
  - **bandwidth (policy-map class)**
  - **fair-queue**
  - **priority**
  - **queue-limit**
  - **random-detect**
  - **shape**
- VASI 2000 pairs are not supported on Open Shortest Path First (OSPF).

- VASI is not supported because Multicast First Hop and Multicast punt packets on VASI interface are not supported.
- Web Cache Communication Protocol (WCCP) is not supported.

Information About Configuring the VRF-Aware Software Infrastructure

VASI Overview

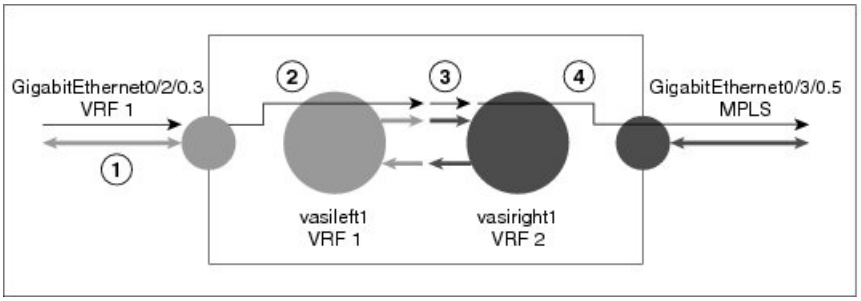
VRF-Aware Software Infrastructure (VASI) provides the ability to apply services such as, a firewall, GETVPN, IPsec, and Network Address Translation (NAT), to traffic that flows across different virtual routing and forwarding (VRF) instances. VASI is implemented by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF instance. The VASI virtual interface is the next-hop interface for any packet that needs to be switched between these two VRF instances. VASI interfaces provide the framework to configure a firewall or NAT between VRF instances.

Each interface pair is associated with two different VRF instances. The pairing is done automatically based on the two interface indexes such that the vasileft interface is automatically paired to the vasiright interface. For example, in the figure below, vasileft1 and vasiright1 are automatically paired, and a packet entering vasileft1 is internally handed over to vasiright1.

On VASI interfaces, you can configure either static routing or dynamic routing with Internal Border Gateway Protocol (IBGP), Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF).

The following figure shows an inter-VRF VASI configuration on the same device.

Figure 73: Inter-VRF VASI Configuration



When an inter-VRF VASI is configured on the same device, the packet flow happens in the following order:

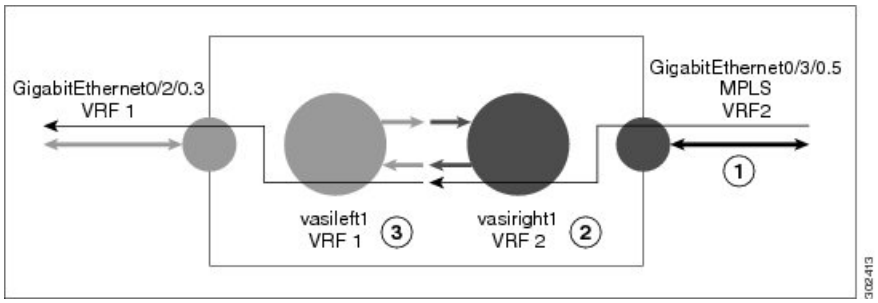
1. A packet enters the physical interface that belongs to VRF 1 (Gigabit Ethernet 0/2/0.3).
2. Before forwarding the packet, a forwarding lookup is done in the VRF 1 routing table. Vasileft1 is chosen as the next hop, and the Time to Live (TTL) value is decremented from the packet. Usually, the forwarding address is selected on the basis of the default route in the VRF. However, the forwarding address can also be a static route or a learned route. The packet is sent to the egress path of vasileft1 and then automatically sent to the vasiright1 ingress path.
3. When the packet enters vasiright1, a forwarding lookup is done in the VRF 2 routing table, and the TTL is decremented again (second time for this packet).

4. VRF 2 forwards the packet to the physical interface, Gigabit Ethernet 0/3/0.5.

The following figure shows how VASI works in a Multiprotocol Label Switching (MPLS) VPN configuration.

Note In the following figure, MPLS is enabled on the Gigabit Ethernet interface, but MPLS traffic is not supported across VASI pairs.

Figure 74: VASI with an MPLS VPN Configuration



When VASI is configured with a Multiprotocol Label Switching (MPLS) VPN, the packet flow happens in the following order:

1. A packet arrives on the MPLS interface with a VPN label.
2. The VPN label is stripped from the packet, a forwarding lookup is done within VRF 2, and the packet is forwarded to vasiright1. The TTL value is decremented from the packet.
3. The packet enters vasileft1 on the ingress path, and another forwarding lookup is done in VRF 1. The packet is sent to the egress physical interface in VRF1 (Gigabit Ethernet 0/2/0.3). The TTL is again decremented from the packet.

Multicast and Multicast VPN on VASI

VRF-Aware Service Infrastructure (VASI) applies services like the zone-based firewall, Network Address Translation (NAT), and IPsec to traffic that travels across different virtual routing and forwarding (VRF) instances. The Multicast and MVPN on VASI feature supports IPv4 and IPv6 multicast and multicast VPN (MVPN) on VASI interfaces. This feature is independent of the multicast modes (sparse, source-specific multicast [SSM] and so on) configured at the customer site and also independent of the MVPN mode—generic routing encapsulation (GRE)-based or Multicast Label Distribution Protocol (MLDP)-based—in the core network.

Multicast reduces traffic in a network by simultaneously delivering a single stream of information to potentially thousands of recipients. Multicast delivers source traffic from an application to multiple receivers without burdening the source or receivers and uses a minimum of network bandwidth. Multicast VPN (MVPN) provides the ability to support multicast over Layer 3 VPNs.

VASI is implemented using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF. VASI virtual interface is the next hop interface for any packet that needs to be switched between these two VRFs. VASI interfaces are virtual interfaces and you can configure IP address and other services like other logical interfaces. You need to enable multicast on VASI interface pairs for this feature to work.

How to Configure the VRF-Aware Software Infrastructure

Configuring a VASI Interface Pair

To configure a VRF-Aware Software Infrastructure (VASI) interface pair, you must configure the **interface vasileft** command on one interface and the **interface vasiright** command on the second interface. The interface numbers must be identical to pair vasileft with vasiright. You can configure a virtual routing and forwarding (VRF) instance on any VASI interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *table-name*
5. **ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*}
6. **exit**
7. **ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number*
8. **interface** *type number*
9. **vrf forwarding** *table-name*
10. **ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*}
11. **exit**
12. **ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number*
13. **end**

DETAILED STEPS

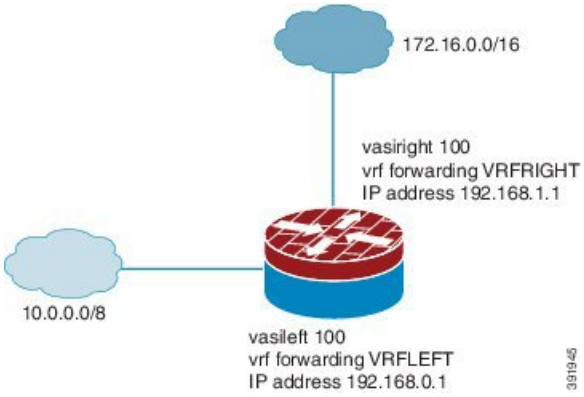
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vasileft 100`	Configures a VASI interface and enters interface configuration mode. • In this example, the vasileft interface is configured.
Step 4	**vrf forwarding** *table-name* **Example:** `Device(config-if)# vrf forwarding VRFLEFT`	Configures a VRF table. **Note** You can configure VRF forwarding on any VASI interface. You need not configure VRF instances on both VASI interfaces.

	Command or Action	Purpose
Step 5	**ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*} **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.255.0`	Configures a primary or secondary IP address for an interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number* **Example:** `Device(config)# ip route vrf VRFLEFT 172.16.0.0 255.255.0.0 VASILEFT 100`	Establishes a static route for a VRF instance and a VASI interface. **Note** To add an IP route for a VRF instance, you must specify the **vrf** keyword.
Step 8	**interface** *type number* **Example:** `Device(config)# interface vasiright 100`	Configures a VASI interface and enters interface configuration mode. • In this example, the vasiright interface is configured.
Step 9	**vrf forwarding** *table-name* **Example:** `Device(config-if)# vrf forwarding VRFRIGHT`	Configures the VRF table.
Step 10	**ip address** {*ip-address mask* [**secondary**] | **pool** *pool-name*} **Example:** `Device(config-if)# ip address 192.168.1.1 255.255.255.0`	Configures a primary or secondary IP address for an interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**ip route** [**vrf** *vrf-name*] *destination-prefix destination-prefix-mask interface-type interface-number* **Example:** `Device(config)# ip route vrf VRFRIGHT 10.0.0.0 255.0.0.0 VASIRIGHT 100`	Establishes a static route for a VRF instance and a VASI interface. **Note** To add an IP route for a VRF instance, you must specify the **vrf** keyword.
Step 13	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for the VRF-Aware Software Infrastructure

Example: Configuring a VASI Interface Pair

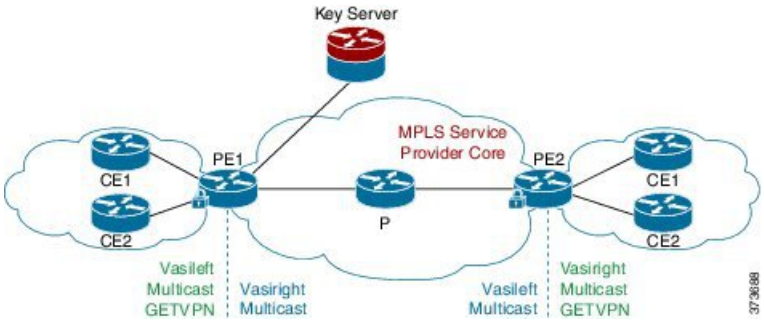
A virtual routing and forwarding (VRF) instance must be enabled for each interface of the VASI pair (VASILEFT and VASIRIGHT). The below example shows how to configure a VASI interface pair.




```
Device(config)# interface vasileft 100
Device(config-if)# vrf forwarding VRFLEFT
Device(config-if)# ip address 192.168.0.1 255.255.255.0
Device(config-if)# exit
Device(config)# ip route vrf VRFLEFT 172.16.0.0 255.255.0.0 vasileft 100
Device(config)# interface vasiright 100
Device(config-if)# vrf forwarding VRFRIGHT
Device(config-if)# ip address 192.168.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# ip route vrf VRFRIGHT 10.0.0.0 255.0.0.0 vasiright 100
Device(config)# end
```

Example: Configuring Multicast and MVPN on VASI

Figure 75: GRE-Based MVPN and GETVPN Configuration



The following example shows how to configure generic routing encapsulation (GRE)-based Multicast VPN (MVPN) and GETVPN on VASI interface pairs. Here, the cryptomap is applied to the vasileft interface. The vasileft interface acts as the customer edge (CE) device and does encryption; the interface is part of the vrf-cust1 virtual routing and forwarding (VRF) instance. The vasiright interface is part of the vrf-core1 VRF instance, to pass traffic across the Multiprotocol Label Switching (MPLS) core and for applied crypto services. The core network supports multicast, and multicast in the VRFs is in stateful switchover (SSO) mode.

```
! PE1 Configuration
Device(config)# vrf definition Mgmt-intf
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# vrf definition vrf-core1
Device(config-vrf)# rd 2:1
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# mdt default 203.0.113.1 ! Enables GRE-based MVPN and mdt default
tree
Device(config-vrf-af)# mdt data 203.0.113.33 255.255.255.224 ! Enables the mdt data tree
Device(config-vrf-af)# route-target export 2:1
Device(config-vrf-af)# route-target import 2:1
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# mdt default 203.0.113.1
Device(config-vrf-af)# mdt data 203.0.113.33 255.255.255.224
Device(config-vrf-af)# route-target export 2:1
Device(config-vrf-af)# route-target import 2:1
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# vrf definition vrf-cust1
Device(config-vrf)# rd 1:1
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
!
Device(config)# logging buffered 10000000
Device(config)# no logging console
!
Device(config)# no aaa new-model
Device(config)# clock timezone CST 8 0
!
Device(config)# ip multicast-routing distributed
Device(config)# ip multicast-routing vrf vrf-core1 distributed
Device(config)# ip multicast-routing vrf vrf-cust1 distributed
!
Device(config)# ipv6 unicast-routing
Device(config)# ipv6 multicast-routing
Device(config)# ipv6 multicast-routing vrf vrf-core1
Device(config)# ipv6 multicast-routing vrf vrf-cust1
!
Device(config)# subscriber templating
Device(config)# mpls label protocol ldp
Device(config)# multilink bundle-name authenticated
Device(config)# spanning-tree extend system-id
!
```

```
Device(config)# cdp run
Device(config)# ip ftp source-interface GigabitEthernet 0
Device(config)# ip tftp source-interface GigabitEthernet 0
Device(config)# ip tftp blocksize 8192
!
Device(config)# class-map match-any maincampus-ratelimit
Device(config-cmap)# match access-group 101
Device(config-cmap)# exit
!
Device(config)# policy-map transit-limt
Device(config-pmap)# description 160mb transit rate limit
Device(config-pmap)# class maincampus-ratelimit
Device(config-pmap-c)# police 160000000 30000000 60000000 conform-action transmit
exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
Device(config-pmap)# exit
!
Device(config)# crypto keyring vrf-cust1 vrf vrf-cust1 ! enables GETVPN
Device(conf-keyring)# pre-shared-key address 0.0.0.0 0.0.0.0 key cisco
Device(conf-keyring)# exit
!
Device(config)# crypto isakmp policy 1
Device(config-isakmp)# encryption 3des
Device(config-isakmp)# authentication pre-share
Device(config-isakmp)# group 2
Device(config-isakmp)# exit
Device(config)# crypto isakmp key cisco address 10.0.3.2
!
Device(config)# crypto gdoi group secure-wan
Device(config-gkm-group)# identity number 12345
Device(config-gkm-group)# server address ipv4 10.0.3.4
Device(config-gkm-group)# exit
!
Device(config)# crypto gdoi group ipv6 ipv6-secure-wan
Device(config-gkm-group)# identity number 123456
Device(config-gkm-group)# server address ipv4 10.0.3.6
Device(config-gkm-group)# exit
!
Device(config)# crypto map getvpn 1 gdoi
Device(config-crypto-map)# set group secure-wan
Device(config-crypto-map)# exit
!
Device(config)# crypto map ipv6 getvpn-v6 1 gdoi
Device(config-crypto-map)# set group ipv6-secure-wan
Device(config-crypto-map)# exit
!
Device(config)# interface loopback 0
Device(config-if)# ip address 198.51.100.241 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address 2001:DB8::1/32
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 192.0.2.1 255.255.255.240
Device(config-if)# shutdown
Device(config-if)# negotiation auto
!
Device(config)# interface GigabitEthernet 0/0/1
```

```
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ip address 192.0.2.18 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# negotiation auto
Device(config-if)# mpls ip
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/1
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 10.0.3.1 255.255.255.0
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/2
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/3
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 192.0.2.34 255.255.255.240
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp version 3
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:0000:0000:0000:0000:0000:0001/48
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0/2/4
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface GigabitEthernet 0
Device(config-if)# vrf forwarding Mgmt-intf
Device(config-if)# ip address 10.74.30.161 255.255.255.0
Device(config-if)# negotiation auto
Device(config-if)# exit
!
Device(config)# interface vasileft 1 ! On the vasileft interface, enable multicast and
GETVPN.
Device(config-if)# vrf forwarding vrf-cust1
Device(config-if)# ip address 209.165.202.129 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address FE80::CEEF:48FF:FEEA:C501 link-local
Device(config-if)# ipv6 address 2001:B000::2/64
Device(config-if)# ipv6 crypto map getvpn-v6
Device(config-if)# ospfv3 100 ipv6 area 0
Device(config-if)# no keepalive
Device(config-if)# crypto map getvpn
Device(config-if)# exit
!
Device(config)# interface vasiright 1 ! On the vasiright interface, only enable multicast.
Device(config-if)# vrf forwarding vrf-core1
Device(config-if)# ip address 209.165.202.130 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ipv6 address 2001:B000::1/64
Device(config-if)# ospfv3 100 ipv6 area 0
```

```
Device(config-if)# no keepalive
Device(config-if)# exit
!
Device(config)# router ospfv3 100
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 unicast vrf vrf-cust1
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 unicast vrf vrf-core1
Device(config-router-af)# redistribute bgp 1
Device(config-router-af)# exit-address-family
!
Device(config)# router ospf 1
Device(config-router)# network 1.1.1.1 0.0.0.0 area 0
Device(config-router)# network 192.0.2.0 0.0.0.255 area 0
Device(config-router)# exit
!
Device(config)# router bgp 1 ! Use BGP routing protocol to broadcast vrf-cust1 routing
entry.
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# neighbor 172.16.0.1 remote-as 1
Device(config-router)# neighbor 172.16.0.1 update-source Loopback0
!
Device(config-router)# address-family ipv4
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 mdt ! For MVPN neighbor setup
Device(config-router-af)# neighbor 172.16.0.1 activate
Device(config-router-af)# neighbor 172.16.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family vpnv6
Device(config-router-af)# neighbor 192.168.0.1 activate
Device(config-router-af)# neighbor 192.168.0.1 send-community both
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 vrf vrf-core1
Device(config-router-af)# bgp router-id 209.165.202.130
Device(config-router-af)# redistribute connected
Device(config-router-af)# neighbor 209.165.202.129 remote-as 65002
Device(config-router-af)# neighbor 209.165.202.129 local-as 65001 no-prepend replace-as
Device(config-router-af)# neighbor 209.165.202.129 activate
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv6 vrf vrf-core1
Device(config-router-af)# redistribute connected
Device(config-router-af)# redistribute ospf 100 include-connected
Device(config-router-af)# bgp router-id 209.165.202.130
Device(config-router-af)# neighbor 2001:B000::2 remote-as 10000
Device(config-router-af)# neighbor 2001:B000::2 local-as 65000 no-prepend replace-as
Device(config-router-af)# neighbor 2001:B000::2 activate
```

```
Device(config-router-af)# exit-address-family
!
Device(config-router)# address-family ipv4 vrf vrf-cust1
Device(config-router-af)# bgp router-id 209.165.202.129
Device(config-router-af)# redistribute connected
Device(config-router-af)# neighbor 209.165.202.130 remote-as 65001
Device(config-router-af)# neighbor 209.165.202.130 local-as 65002 no-prepend replace-as
Device(config-router-af)# neighbor 209.165.202.130 activate
Device(config-router-af)# exit-address-family
Device(config-router)# exit
!
Device(config-router)# address-family ipv6 vrf vrf-cust1
Device(config-router-af)# redistribute connected
Device(config-router-af)# redistribute ospf 100 include-connected
Device(config-router-af)# bgp router-id 209.165.202.129
Device(config-router-af)# neighbor 2001:B000::1 remote-as 65000
Device(config-router-af)# neighbor 2001:B000::1 local-as 10000 no-prepend replace-as
Device(config-router-af)# neighbor 2001:B000::1 activate
Device(config-router-af)# exit-address-family
!
Device(config)# ip forward-protocol nd
!
Device(config)# no ip http server
Device(config)# no ip http secure-server
Device(config)# ip pim rp-address 1.1.1.1
Device(config)# ip pim vrf vrf-core1 ssm default
Device(config)# ip pim vrf vrf-cust1 ssm default
Device(config)# ip route 192.0.2.0 255.255.255.240 10.11.12.10
Device(config)# ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.74.9.1
!
Device(config)# ip access-list standard bidir
Device(config-std-nacl)# exit
!
Device(config)# access-list 101 deny ip 198.51.100.1 255.255.255.240 198.51.100.177
255.255.255.240
Device(config)# ipv6 router eigrp 300
Device(config-rtr)# passive-interface Loopback 0
Device(config-rtr)# redistribute connected
Device(config-rtr)# exit
!
Device(config)# mpls ldp router-id Loopback 0
Device(config)# control-plane
Device(config-cp)# exit
!
Device(config)# line con 0
Device(config-line)# exec-timeout 0 0
Device(config-line)# privilege level 15
Device(config-line)# logging synchronous
Device(config-line)# stopbits 1
Device(config-line)# exit
Device(config)# line vty 0 4
Device(config-line)# exec-timeout 0 0
Device(config-line)# privilege level 15
Device(config-line)# logging synchronous
Device(config-line)# no login
Device(config-line)# end
```

Verifying Multicast VASI Configuration

Use the following commands to verify the multicast VRF-Aware Software Infrastructure (VASI) configuration:

SUMMARY STEPS

1. **enable**
2. **show ip mroute**
3. **show ip mroute vrf**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 show ip mroute

Displays the contents of the multicast routing (mroute) table.

Example:

```
Device# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 203.0.113.1), 04:33:39/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0/2, Forward/Sparse-Dense, 04:33:39/stopped
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:33:39/stopped
(10.0.0.3, 203.0.113.1), 04:33:36/00:00:36, flags: T
  Incoming interface: GigabitEthernet0/0/2, RPF nbr 10.1.1.3
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:33:36/stopped
(10.0.0.1, 203.0.113.1), 04:33:39/00:02:44, flags: T
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 10.1.1.0
  Outgoing interface list:
    GigabitEthernet0/0/2, Forward/Sparse-Dense, 04:33:39/stopped
```

Step 3 **show ip mroute vrf**

Filters the output to display only the contents of the multicast routing table that pertains to the Multicast VPN (MVPN) routing and forwarding (MVRF) instance specified for the *vrf-name* argument.

Example:

```
Device# show ip mroute vrf cust1

(10.2.1.1, 203.1.113.4), 00:40:09/00:02:44, flags: sTI
  Incoming interface: vasileft1, RPF nbr 36.1.1.2
  Outgoing interface list:
    GigabitEthernet0/0/1.1, Forward/Sparse-Dense, 00:40:09/00:02:44
PE1#sh ip mroute vrf cust1-core
(10.2.1.1, 203.1.113.4), 04:22:09/00:02:50, flags: sT
  Incoming interface: Tunnel0, RPF nbr 10.0.0.3
  Outgoing interface list:
    vasiright1, Forward/Sparse-Dense, 04:22:09/00:02:50
PE1#sh ip mroute
(*, 203.1.113.4), 21:08:36/stopped, RP 0.0.0.0, flags: DCZ
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 04:27:50/stopped
    MVRF cust1-core, Forward/Sparse-Dense, 21:06:53/stopped
(10.0.0.3, 203.1.113.4), 04:26:53/00:01:22, flags: TZ
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 10.1.1.1
  Outgoing interface list:
    MVRF cust1-core, Forward/Sparse-Dense, 04:26:53/stopped
```

Additional References for Configuring the VRF-Aware Software Infrastructure

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring the VRF-Aware Software Infrastructure

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 183: Feature Information for Configuring the VRF-Aware Software Infrastructure

Feature Name	Releases	Feature Information
Multicast and Multicast VPN on VASI	Cisco IOS XE Release 3.14S	The Multicast and MVPN on VASI feature supports IPv4 and IPv6 multicast and multicast VPN (MVPN) on VASI interfaces. This feature is independent of the multicast modes (sparse, source-specific multicast [SSM] and so on) configured at the customer site and also independent of the MVPN mode—generic routing encapsulation (GRE)-based or Multicast Label Distribution Protocol (MLDP)-based—in the core network. No new commands have been introduced or modified for this feature.
VRF-Aware Software Infrastructure	Cisco IOS XE Release 2.6	The VRF-Aware Software Infrastructure feature allows you to apply services such as ACLs, NAT, policing, and zone-based firewalls to traffic that flows across two different VRF instances. The VRF-Aware Software Infrastructure (VASI) interfaces support redundancy of the RP and FP. This feature supports IPv4 and IPv6 unicast and multicast traffic on VASI interfaces.

Feature Name	Releases	Feature Information
VASI (VRF-Aware Software Infrastructure) Enhancements Phase I	Cisco IOS XE Release 3.1S	The VASI Enhancements Phase I feature provides the following enhancements to VASI: • Support for 500 VASI interfaces. • Support for IBGP dynamic routing between VASI interfaces.
VASI (VRF-Aware Software Infrastructure) Enhancements Phase II	Cisco IOS XE Release 3.2S	The VASI Enhancements Phase II feature provides the following enhancements to VASI: • Support for IPv6 unicast traffic over VASI interfaces. • Support for OSPF and EIGRP dynamic routing between VASI interfaces.
VASI (VRF-Aware Software Infrastructure) Scale	Cisco IOS XE Release 3.3S	The VASI Scale feature provides support for 1000 VASI interfaces. The following command was introduced or modified: **interface (VASI)**.
VASI (VRF-Aware Software Infrastructure) Scale	Cisco IOS XE Release 3.7.2S	The VASI Scale feature provides support for eBGP dynamic routing between VASI interfaces.
VASI 2000 Pair Scale	Cisco IOS XE Release 3.10S	The VASI 2000 Pair Scale feature provides support for 2000 VASI interfaces. 2000 VASI interfaces are supported on Border Gateway Protocol (BGP). The following command was introduced or modified: **interface (VASI)**.

CHAPTER 138

FTP66 ALG Support for IPv6 Firewalls

The FTP66 ALG Support for IPv6 Firewalls feature allows FTP to work with IPv6 firewalls. This module describes how to configure a firewall, Network Address Translation (NAT), and Stateful NAT64 to work with the FTP66 application-level gateway (ALG).

Restrictions for FTP66 ALG Support for IPv6 Firewalls

The FTP66 ALG does not support the following:

- Box-to-box high availability.
- Per-subscriber firewalls.
- Stateless Network Address Translation 64 (NAT64).
- Virtual routing and forwarding (VRF) when stateful NAT64 is configured.
- Virtual TCP (vTCP) or the breaking up of packets into smaller packets after translation.

Information About FTP66 ALG Support for IPv6 Firewalls

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.

- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

FTP66 ALG Support Overview

Firewalls support the inspection of IPv6 packets and stateful Network Address Translation 64 (NAT64). For FTP to work over IPv6 packet inspection, the application-layer gateway (ALG) (also called the application-level gateway [ALG]), FTP66, is required. The FTP66 ALG is also called all-in-one FTP ALG and one FTP ALG.

The FTP66 ALG supports the following:

- Firewall IPv4 packet inspection
- Firewall IPv6 packet inspection
- NAT configuration
- NAT64 configuration (along with FTP64 support)
- NAT and firewall configuration
- NAT64 and firewall configuration

The FTP66 ALG has the following security vulnerabilities:

- Packet segmentation attack—The FTP ALG state machine can detect segmented packets, and the state machine processing is stopped until a complete packet is received.
- Bounce attack—The FTP ALG does not create doors (for NAT) or pinholes (for firewalls) with a data port number less than 1024. The prevention of a bounce attack is activated only when the firewall is enabled.

FTP Commands Supported by FTP66 ALG

The FTP66 application-level gateway (ALG) is based on RFC 959. This section describes the main RFC 959 and RFC 2428 FTP commands and responses that the FTP66 ALG processes.

PORT Command

The PORT command is used in active FTP mode. The PORT command specifies the address and the port number to which a server should connect. When you use this command, the argument is a concatenation of a 32-bit Internet host address and a 16-bit TCP port address. The address information is broken into 8-bit fields, and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas.

The following is a sample PORT command, where *h1* is the highest order 8-bit of the Internet host address:

```
PORT h1,h2,h3,h4,p1,p2
```

PASV Command

The PASV command requests a server to listen on a data port that is not the default data port of the server and to wait for a connection, rather than initiate another connection, when a TRANSFER command is received. The response to the PASV command includes the host and port address the server is listening on.

Extended FTP Commands

Extended FTP commands provide a method by which FTP can communicate the data connection endpoint information for network protocols other than IPv4. Extended FTP commands are specified in RFC 2428. In RFC 2428, the extended FTP commands EPRT and EPSV, replace the FTP commands PORT and PASV, respectively.

EPRT Command

The EPRT command allows you to specify an extended address for data connection. The extended address must consist of a network protocol, network address, and transport address. The format of an EPRT command is as follows:

```
EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
```

- The <net-prt> argument must be an address family number and must be defined as described in the table below.

Table 184: The <net-prt> Argument Definitions

Address Family Number	Protocol
1	IPv4 (Pos81a)
2	IPv6 (DH96)

- The <net-addr> argument is a protocol-specific string representation of the network address. For the two address family numbers specified in the table above (address family numbers 1 and 2), the addresses must be in the format listed in the table below.

Address Family Number	Address Format	Example
1	Dotted decimal	10.135.1.2
2	IPv6 string representations defined in DH96	2001:DB8:1::1

- The <tcp-port> argument must be a string representation of the number of the TCP port on which the host is listening for data connection.
- The following command shows how to specify the server to use an IPv4 address to open a data connection to host 10.235.1.2 on TCP port 6275:

```
EPRT |1|10.235.1.2|6275|
```

- The following command shows how to specify the server to use an IPv6 network protocol and a network address to open a TCP data connection on port 5282:

```
EPRT |2|2001:DB8:2::2:417A|5282|
```

- The <d> argument is the delimiter character and it must be in ASCII format, in the range from 33 to 126.

EPSV Command

The EPSV command requests that a server listen on a data port and wait for a connection. The response to this command includes only the TCP port number of the listening connection. The response code for entering passive mode by using an extended address must be 229.

The text returned in response to an EPSV command must be in the following format:

```
(<d><d><d><tcp-port><d>)
```

- The portion of the string enclosed in parentheses must be the exact string needed by the EPRT command to open the data connection.

  The first two fields in parentheses must be blank. The third field must be a string representation of the TCP port number on which the server is listening for a data connection. The network protocol used by the data connection is the same network protocol used by the control connection. The network address used to establish the data connection is the same network address used for the control connection.

- The following is a sample response string:

  ```
  Entering Extended Passive Mode (|||6446|)
  ```

The following FTP responses and commands are also processed by the FTP66 ALG. The results of processing these commands are used to drive the transition in the state machine.

- 230 response
- AUTH
- USER
- PASS

How to Configure FTP66 ALG Support for IPv6 Firewalls

Configuring a Firewall for FTP66 ALG Support

You need to explicitly enable the FTP66 ALG by using the **match protocol ftp** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **exit**

12. **exit**
13. **zone security** *zone-name*
14. **exit**
15. **zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone*
16. **service-policy type inspect** *policy-map-name*
17. **exit**
18. **interface** *type number*
19. **no ip address**
20. **ip virtual-reassembly**
21. **zone-member security** *zone-name*
22. **negotiation auto**
23. **ipv6 address** *ipv6-address/prefix-length*
24. **cdp enable**
25. **exit**
26. **ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number*
27. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
28. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any in2out-class`	Creates an inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol ftp`	Configures a match criteria for a class map on the basis of the named protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect in-to-out`	Creates an inspect type policy map and enters QoS policy-map configuration mode.

	Command or Action	Purpose
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect in2out-class`	Specifies the class on which an action is performed and enters QoS policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class and enters QoS policy-map class configuration mode. • If the traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 12	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 13	**zone security** *zone-name* **Example:** `Device(config)# zone security inside`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 14	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 15	**zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security in2out source inside destination outside`	Creates a pair of security zones and enters security zone-pair configuration mode. • To apply a policy, you must configure a zone pair.

	Command or Action	Purpose
Step 16	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect in-to-out`	Attaches a firewall policy map to the destination zone pair. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 17	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 18	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 19	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 20	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables virtual fragmentation reassembly (VFR) on an interface.
Step 21	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 22	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 23	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 24	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables Cisco Discovery Protocol on an interface.
Step 25	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 26	**ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001::/96 gigabitethernet 0/0/1`	Establishes static IPv6 routes.
Step 27	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::1 gigabitethernet 0/0/1 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.
Step 28	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring NAT for FTP66 ALG Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat inside**
6. **zone-member security** *zone-name*
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **zone-member security** *zone-name*
12. **exit**
13. **ip nat inside source static** *local-ip global-ip*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/2`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that an interface is connected to the inside network (the network that is subject to NAT translation).
Step 6	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.2.1.1 255.255.255.0`	Indicates that an interface is connected to the inside network (the network that is subject to NAT translation).
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.

	Command or Action	Purpose
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security outside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 13	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 10.1.1.10 10.1.1.80`	Enables NAT of the inside source address.
Step 14	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring NAT64 for FTP66 ALG Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no ip address**
6. **ipv6 virtual-reassembly**
7. **zone-member security** *zone-name*
8. **negotiation auto**
9. **ipv6 address** *ipv6-address*
10. **ipv6 enable**
11. **nat64 enable**
12. **cdp enable**
13. **exit**
14. **interface** *type number*
15. **ip address** *type number*
16. **ip virtual-reassembly**
17. **zone member security** *zone-name*

18. **negotiation auto**
19. **nat64 enable**
20. **exit**
21. **ipv6 route** *ipv6-address interface-type interface-number*
22. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
23. **nat64 v6v4 static** *ipv6-address ipv4-address*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes an IP address or disables IP processing.
Step 6	**ipv6 virtual-reassembly** **Example:** `Device(config-if)# ipv6 virtual-reassembly`	Enables virtual fragmentation reassembly (VFR) on an interface.
Step 7	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 8	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 9	**ipv6 address** *ipv6-address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::2/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 10	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 11	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 12	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables Cisco Discovery Protocol on an interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 15	**ip address** *type number* **Example:** `Device(config-if)# ip address 209.165.201.25 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 16	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables VFR on an interface.
Step 17	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security outside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair

	Command or Action	Purpose
		to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 18	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 19	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 on an interface.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 21	**ipv6 route** *ipv6-address interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0`	Establishes static IPv6 routes and specifies the IPv6 address of the next hop that can be used to reach a specified network.
Step 22	**ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address* **Example:** `Device(config)# ipv6 neighbor 2001:DB8:1::103 gigabitethernet 0/0/0 0000.29f1.4841`	Configures a static entry in the IPv6 neighbor discovery cache.
Step 23	**nat64 v6v4 static** *ipv6-address ipv4-address* **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::103 209.165.201.32`	Translates an IPv6 source address to an IPv4 source address and an IPv4 destination address to an IPv6 destination address for NAT64.
Step 24	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for FTP66 ALG Support for IPv6 Firewalls

Example: Configuring an IPv6 Firewall for FTP66 ALG Support

```
Device# configure terminal
Device(config)# class-map type inspect match-any in2out-class
Device(config-cmap)# match protocol ftp
Device(config-cmap)# exit
Device(config)# policy-map type inspect in-to-out
```

```
Device(config-pmap)# class type inspect in2out-class
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone-pair security in2out source inside destination outside
Device(config-sec-zone-pair)# service-policy type inspect in-to-out
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone-member security inside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(config-if)# cdp enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# no ip address
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone-member security outside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:2::2/96
Device(config-if)# exit
Device(config)# ipv6 route 2001::/96 gigabitethernet 0/0/1
Device(config)# ipv6 route 2001::/96 gigabitethernet 0/1/1
Device(config)# ipv6 neighbor 2001:DB8:1::1 gigabitethernet 0/0/1 0000.29f1.4841
Device(config)# ipv6 neighbor 2001:DB8:2::2 gigabitethernet 0/1/1 0000.29f1.4842
Device(config)# end
```

Example: Configuring NAT for FTP66 ALG Support

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1/2
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip address 10.2.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# zone-member security outside
Device(config-if)# exit
Device(config-if)# ip nat inside source static 10.1.1.10 10.1.1.80
```

Example: Configuring NAT64 for FTP66 ALG Support

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no ip address
Device(config-if)# ipv6 virtual-reassembly
```

```
Device(config-if)# zone-member security inside
Device(config-if)# negotiation auto
Device(config-if)# ipv6 address 2001:DB8:1::2/96
Device(config-if)# ipv6 enable
Device(config-if)# nat64 enable
Device(config-if)# cdp enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip address 209.165.201.25 255.255.255.0
Device(config-if)# ip virtual-reassembly
Device(config-if)# zone member security outside
Device(config-if)# negotiation auto
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0
Device(config)# 2001:DB8:1::103 gigabitethernet 0/0/0 0000.29f1.4841
Device(config)# nat64 v6v4 static 2001:DB8:1::103 209.165.201.32
```

Additional References for FTP66 ALG Support for IPv6 Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
NAT commands	IP Addressing Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 959	*File Transfer Protocol*
RFC 2428	*FTP Extensions for IPv6 and NATs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for FTP66 ALG Support for IPv6 Firewalls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 185: Feature Information for FTP66 ALG Support for IPv6 Firewalls

Feature Name	Releases	Feature Information
FTP66 ALG Support for IPv6 Firewalls	Cisco IOS XE Release 3.7S	The FTP66 ALG Support for IPv6 Firewalls feature allows FTP to work with IPv6 firewalls. This module describes how to configure a firewall, Network Address Translation (NAT), and NAT64 to work with the FTP66 application-level gateway (ALG).

CHAPTER 139

Protection Against Distributed Denial of Service Attacks

The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. In Cisco IOS XE Release 3.4S and later releases, you can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, the half-opened connections limit, and global TCP SYN cookie protection to prevent distributed DoS attacks.

Information About Protection Against Distributed Denial of Service Attacks

Aggressive Aging of Firewall Sessions

The Aggressive Aging feature provides the firewall the capability of aggressively aging out sessions to make room for new sessions, thereby protecting the firewall session database from filling. The firewall protects its resources by removing idle sessions. The Aggressive Aging feature allows firewall sessions to exist for a shorter period of time defined by a timer called aging-out time.

The Aggressive Aging feature includes thresholds to define the start and end of the aggressive aging period—high and low watermarks. The aggressive aging period starts when the session table crosses the high watermark and ends when it falls below the low watermark. During the aggressive aging period, sessions will exist for a shorter period of time that you have configured by using the aging-out time. If an attacker initiates sessions at a rate that is faster than the rate at which the firewall terminates sessions, all resources that are allocated for creating sessions are used and all new connections are rejected. To prevent such attacks, you can configure the Aggressive Aging feature to aggressively age out sessions. This feature is disabled by default.

You can configure aggressive aging for half-opened sessions and total sessions at the box level (box refers to the entire firewall session table) and the virtual routing and forwarding (VRF) level. If you have configured

this feature for total sessions, all sessions that consume firewall session resources are taken into account. Total sessions comprise established sessions, half-opened sessions, and sessions in the imprecise session database. (A TCP session that has not yet reached the established state is called a half-opened session.)

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (the source IP address, the destination IP address, the source port, the destination port, and the protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on). In the case of aggressive aging for half-opened sessions, only half-opened sessions are considered.

You can configure an aggressive aging-out time for Internet Control Message Protocol (ICMP), TCP, and UDP firewall sessions. The aging-out time is set by default to the idle time.

Event Rate Monitoring Feature

The Event Rate Monitoring feature monitors the rate of predefined events in a zone. The Event Rate Monitoring feature includes basic threat detection, which is the ability of a security device to detect possible threats, anomalies, and attacks to resources inside the firewall and to take action against them. You can configure a basic threat detection rate for events. When the incoming rate of a certain type of event exceeds the configured threat detection rate, event rate monitoring considers this event as a threat and takes action to stop the threat. Threat detection inspects events only on the ingress zone (if the Event Rate Monitoring feature is enabled on the ingress zone).

The network administrator is informed about the potential threats via an alert message (syslog or high-speed logger [HSL]) and can take actions such as detecting the attack vector, detecting the zone from which the attack is coming, or configuring devices in the network to block certain behaviors or traffic.

The Event Rate Monitoring feature monitors the following types of events:

- Firewall drops due to basic firewall checks failure—This can include zone or zone-pair check failures, or firewall policies configured with the drop action, and so on.
- Firewall drops due to Layer 4 inspection failure—This can include TCP inspections that have failed because the first TCP packet is not a synchronization (SYN) packet.
- TCP SYN cookie attack—This can include counting the number of SYN packets that are dropped and the number of SYN cookies that are sent as a spoofing attack.

The Event Rate Monitoring feature monitors the average rate and the burst rate of different events. Each event type has a rate object that is controlled by an associated rate that has a configurable parameter set (the average threshold, the burst threshold, and a time period). The time period is divided into time slots; each time slot is 1/30th of the time period.

The average rate is calculated for every event type. Each rate object holds 30 completed sampling values plus one value to hold the current ongoing sampling period. The current sampling value replaces the oldest calculated value and the average is recalculated. The average rate is calculated during every time period. If the average rate exceeds the average threshold, the Event Rate Monitoring feature will consider this as a possible threat, update the statistics, and inform the network administrator.

The burst rate is implemented by using the token bucket algorithm. For each time slot, the token bucket is filled with tokens. For each event that occurs (of a specific event type), a token is removed from the bucket. An empty bucket means that the burst threshold is reached, and the administrator receives an alarm through the syslog or HSL. You can view the threat detection statistics and learn about possible threats to various events in the zone from the output of the **show policy-firewall stats zone** command.

You must first enable basic threat detection by using the **threat-detection basic-threat** command. Once basic threat detection is configured, you can configure the threat detection rate. To configure the threat detection rate, use the **threat-detection rate** command.

The following table describes the basic threat detection default settings that are applicable if the Event Rate Monitoring feature is enabled.

Table 186: Basic Threat Detection Default Settings

Packet Drop Reason	Threat Detection Settings
Basic firewall drops	average-rate 400 packets per second (pps) burst-rate 1600 pps rate-interval 600 seconds
Inspection-based firewall drops	average-rate 400 pps burst-rate 1600 pps rate-interval 600 seconds
SYN attack firewall drops	average-rate 100 pps burst-rate 200 pps rate-interval 600 seconds

Half-Opened Connections Limit

The firewall session table supports the limiting of half-opened firewall connections. Limiting the number of half-opened sessions will defend the firewall against attacks that might fill the firewall session table at the per-box level or at the virtual routing and forwarding (VRF) level with half-opened sessions and prevent sessions from being established. The half-opened connection limit can be configured for Layer 4 protocols, Internet Control Message Protocol (ICMP), TCP, and UDP. The limit set to the number of UDP half-opened sessions will not affect the TCP or ICMP half-opened sessions. When the configured half-opened session limit is exceeded, all new sessions are rejected and a log message is generated, either in syslog or in the high-speed logger (HSL).

The following sessions are considered as half-opened sessions:

- TCP sessions that have not completed the three-way handshake.
- UDP sessions that have only one packet detected in the UDP flow.
- ICMP sessions that do not receive a reply to the ICMP echo request or the ICMP time-stamp request.

TCP SYN-Flood Attacks

You can configure the global TCP SYN-flood limit to limit SYN flood attacks. TCP SYN-flooding attacks are a type of denial of service (DoS) attack. When the configured TCP SYN-flood limit is reached, the firewall verifies the source of sessions before creating more sessions. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or program tries to use false data to gain access to resources

in a network. TCP SYN flooding can take up all resources on a firewall or an end host, thereby causing denial of service to legitimate traffic. You can configure TCP SYN-flood protection at the VRF level and the zone level.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host intending to utilize all resources on that host.
- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the intention of exhausting the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall.

How to Configure Protection Against Distributed Denial of Service Attacks

Configuring a Firewall

In this task, you will do the following:

- Configure a firewall.
- Create a security source zone.
- Create a security destination zone.
- Create a security zone pair by using the configured source and destination zones.
- Configure an interface as a zone member.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** {**icmp** | **tcp** | **udp**}
5. **exit**
6. **parameter-map type inspect global**
7. **redundancy**
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect**
12. **exit**
13. **class class-default**
14. **drop**
15. **exit**
16. **exit**
17. **zone security** *security-zone-name*

18. **exit**
19. **zone security** *security-zone-name*
20. **exit**
21. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
22. **service-policy type inspect** *policy-map-name*
23. **exit**
24. **interface** *type number*
25. **ip address** *ip-address mask*
26. **encapsulation dot1q** *vlan-id*
27. **zone-member security** *security-zone-name*
28. **end**
29. To attach a zone to another interface, repeat Steps 21 to 25.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any ddos-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** {**icmp** | **tcp** | **udp**} **Example:** `Device(config-cmap)# match protocol tcp`	Configures the match criterion for a class map based on the specified protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 7	**redundancy** **Example:** `Device(config-profile)# redundancy`	Enables firewall high availability.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 11	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 13	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Configures the default class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 14	**drop** **Example:** `Device(config-pmap-c)# drop`	Allows traffic to pass between two interfaces in the same zone.
Step 15	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters QoS policy-map configuration mode.
Step 16	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 17	**zone security** *security-zone-name* **Example:** `Device(config)# zone security private`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair—a source and a destination zone.
Step 18	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 19	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair—a source and a destination zone.
Step 20	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 21	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private2public source private destination public`	Creates a zone pair and enters security zone-pair configuration mode.
Step 22	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect ddos-fw`	Attaches a policy map to a top-level policy map.
Step 23	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 24	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0.1`	Configures an interface and enters subinterface configuration mode.
Step 25	**ip address** *ip-address mask* **Example:** `Device(config-subif)# ip address 10.1.1.1 255.255.255.0`	Configures an IP address for the subinterface.
Step 26	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 2`	Sets the encapsulation method used by the interface.
Step 27	**zone-member security** *security-zone-name* **Example:** `Device(config-subif)# zone-member security private`	Configures the interface as a zone member. • For the *security-zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is

	Command or Action	Purpose
		a zone member, you must make that zone part of a zone pair to which you apply a policy. If the policy permits traffic (via **inspect** or **pass** actions), traffic can flow through the interface.
Step 28	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and enters privileged EXEC mode.
Step 29	To attach a zone to another interface, repeat Steps 21 to 25.	—

Configuring the Aggressive Aging of Firewall Sessions

You can configure the Aggressive Aging feature for per-box (per-box refers to the entire firewall session table), default-VRF, and per-VRF firewall sessions. Before the Aggressive Aging feature can work, you must configure the aggressive aging and the aging-out time of firewall sessions.

Perform the following tasks to configure the aggressive aging of firewall sessions.

Configuring per-Box Aggressive Aging

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
5. **per-box aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Steps 4 and 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging rate for half-opened sessions in the firewall session table.
Step 5	**per-box aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box aggressive-aging high 1700 low 1300`	Configures the aggressive aging limit of total sessions.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:**	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.

	Command or Action	Purpose
	`Device(config-profile)# tcp synwait-time 30 ageout-time 10`	• After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring Aggressive Aging for a Default VRF

When you configure the **max-incomplete aggressive-aging** `command, it applies to the default VRF.`

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enters one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
5. **session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}]
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enters one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255`	Configures the maximum limit and the aggressive aging limit of half-opened firewall sessions.
Step 5	**session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total limit and the aggressive aging limit for total firewall sessions.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example,

	Command or Action	Purpose
		instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	Displays global VRF firewall policy statistics.

Configuring the Aging Out of Firewall Sessions

You can configure the aging out of ICMP, TCP, or UDP firewall sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
5. **exit**
6. **parameter-map type inspect** *parameter-map-name*
7. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **exit**
10. **policy-map type inspect** *policy-map-name*
11. **class type inspect match-any** *class-map-name*
12. **inspect** *parameter-map-name*
13. **end**
14. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspectglobal`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 4 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 7	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions. • You can also configure the **tcp finwait-time** command to specify how long a TCP session will be managed after the firewall detects a finish (FIN) exchange, or you can configure the **tcp synwait-time** command to specify how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the

	Command or Action	Purpose
		default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is enabled when the connections drop below the low watermark.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 11	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 12	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 14	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

 VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
       Total Session Count(estab + half-open): 270, Exceed: 0
       Total Session Aggressive Aging Period Off, Event Count: 0

                Half Open
       Protocol Session Cnt     Exceed
       -------- -----------     ------
       All      0               0
       UDP      0               0
       ICMP     0               0
       TCP      0               0
```

```
TCP Syn Flood Half Open Count: 0, Exceed: 12
Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring per-VRF Aggressive Aging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **parameter-map type inspect-vrf** *vrf-pmap-name*
9. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
10. **session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}]
11. **alert on**
12. **exit**
13. Enter one of the following commands:
    - **parameter-map type inspect-global**
    - **parameter-map type inspect global**
14. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
15. **exit**
16. **parameter-map type inspect** *parameter-map-name*
17. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
18. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
19. **exit**
20. **policy-map type inspect** *policy-map-name*
21. **class type inspect match-any** *class-map-name*
22. **inspect** *parameter-map-name*
23. **end**
24. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf ddos-vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:2`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:2`	Creates a route-target extended community and exports the routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:2`	Creates a route-target extended community and imports routing information from the target VPN extended community.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 8	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 9	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging limit for half-opened sessions.
Step 10	**session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total session limit and the aggressive aging limit for the total sessions. • You can configure the total session limit as an absolute value or as a percentage.
Step 11	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 13	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 14 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 14	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 15	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 16	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 17	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions.
Step 18	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.

	Command or Action	Purpose
Step 19	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 20	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 21	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 22	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 23	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 24	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
       Total Session Count(estab + half-open): 80, Exceed: 0
       Total Session Aggressive Aging Period Off, Event Count: 0

                Half Open
       Protocol Session Cnt     Exceed
       -------- -----------     ------
       All      0               0
       UDP      0               0
       ICMP     0               0
       TCP      0               0

       TCP Syn Flood Half Open Count: 0, Exceed: 116
       Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring Firewall Event Rate Monitoring

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **alert on**
5. **threat-detection basic-threat**
6. **threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
7. **threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
8. **threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
9. **exit**
10. **zone security** *security-zone-name*
11. **protection** *parameter-map-name*
12. **exit**
13. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
14. **end**
15. **show policy-firewall stats zone**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-zone** *zone-pmap-name* **Example:** `Device(config)# parameter-map type inspect-zone zone-pmap1`	Configures an inspect-zone parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages for a zone. • You can use the **log** command to configure the logging of alerts either to the syslog or to the high-speed logger (HSL).

	Command or Action	Purpose
Step 5	**threat-detection basic-threat** **Example:** `Device(config-profile)# threat-detection basic-threat`	Configures basic threat detection for a zone.
Step 6	**threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall drop events. • You must configure the **threat-detection basic-threat** command before you configure the **threat-detection rate** command.
Step 7	**threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall inspection-based drop events.
Step 8	**threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate syn-attack average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for TCP SYN attack events.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode.
Step 11	**protection** *parameter-map-name* **Example:** `Device(config-sec-zone)# protection zone-pmap1`	Attaches the inspect-zone parameter map to the zone and applies the features configured in the inspect-zone parameter map to the zone.
Step 12	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 13	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private2public source private destination public`	Creates a zone pair and enters security zone-pair configuration mode.
Step 14	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.
Step 15	**show policy-firewall stats zone** **Example:** `Device# show policy-firewall stats zone`	Displays policy firewall statistics at the zone level.

Configuring the per-Box Half-Opened Session Limit

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **alert on**
5. **per-box max-incomplete** *number*
6. **session total** *number*
7. **end**
8. **show policy-firewall stats global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip to Steps 5 and 6 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box max-incomplete** *number* **Example:** `Device(config-profile)# per-box max-incomplete 12345`	Configures the maximum number of half-opened connections for the firewall session table.
Step 6	**session total** *number* **Example:** `Device(config-profile)# session total 34500`	Configures the total session limit for the firewall session table.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 8	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring the Half-Opened Session Limit for an Inspect-VRF Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-name*

4. **alert on**
5. **max-incomplete** *number*
6. **session total** *number*
7. **exit**
8. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
9. **alert on**
10. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
11. **end**
12. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect-VRF parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**max-incomplete** *number* **Example:** `Device(config-profile)# max-incomplete 2000`	Configures the maximum number of half-opened connections per VRF.
Step 6	**session total** *number* **Example:** `Device(config-profile)# session total 34500`	Configures the total session limit for a VRF.
Step 7	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

	Command or Action	Purpose

Step 8	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, you can use either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 10 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 9	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 10	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds the VRF to the global parameter map.
Step 11	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 12	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Configuring the Global TCP SYN Flood Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **alert on**
5. **per-box tcp syn-flood limit** *number*

6. **end**
7. **show policy-firewall stats vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, you can configure either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box tcp syn-flood limit** *number* **Example:** `Device(config-profile)# per-box tcp syn-flood limit 500`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 7	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-opened sessions are present.

Example

The following is sample output from the **show policy-firewall stats vrf global** command:

```
Device# show policy-firewall stats vrf global

Global table statistics
       total_session_cnt: 0
       exceed_cnt:        0
       tcp_half_open_cnt: 0
       syn_exceed_cnt:    0
```

Configuration Examples for Protection Against Distributed Denial of Service Attacks

Example: Configuring a Firewall

```
Router# configure terminal
Router(config)# class-map type inspect match-any ddos-class
Router(config-cmap)# match protocol tcp
Router(config-cmap-c)# exit
Router(config)# parameter-map type inspect global
Router(config-profile)# redundancy
Router(config-profile)# exit
Router(config)# policy-map type inspect ddos-fw
Router(config-pmap)# class type inspect ddos-class
Router(config-pmap-c)# inspect
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# zone security private
Router(config-sec-zone)# exit
Router(config)# zone security public
Router(config-sec-zone)# exit
Router(config)# zone-pair security private2public source private destination public
Router((config-sec-zone-pair)# service-policy type inspect ddos-fw
Router((config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/1/0.1
Router(config-subif)# ip address 10.1.1.1 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# zone-member security private
Router(config-subif)# exit
Router(config)# interface gigabitethernet 1/1/0.1
Router(config-subif)# ip address 10.2.2.2 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# zone-member security public
Router(config-subif)# end
```

Example: Configuring the Aggressive Aging of Firewall Sessions

Example: Configuring per-Box Aggressive Aging

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# per-box max-incomplete 2000 aggressive-aging 1500 low 1200
Device(config-profile)# per-box aggressive-aging high 1700 low 1300
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring Aggressive Aging for a Default VRF

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring the Aging Out of Firewall Sessions

```
Device# configure terminal
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-profile)# class type inspect match-any ddos-class
Device(config-profile)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring per-VRF Aggressive Aging

```
Device# configure terminal
Device(config)# ip vrf ddos-vrf1
Device(config-vrf)# rd 100:2
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# exit
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# alert on
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
```

```
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-pmap)# class type inspect match-any ddos-class
Device(config-pmap-c)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring Firewall Event Rate Monitoring

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect zone zone-pmap1
Device(config-profile)# alert on
Device(config-profile)# threat-detection basic-threat
Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold
 100 burst-threshold 100
Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# threat-detection rate syn-attack average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# exit
Device(config)# zone security public
Device(config-sec-zone)# protection zone-pmap1
Device(config-sec-zone)# exit
Device(config)# zone-pair security private2public source private destination public
Device(config-sec-zone-pair)# end
```

Example: Configuring the per-Box Half-Opened Session Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box max-incomplete 12345
Device(config-profile)# session total 34500
Device(config-profile)# end
```

Example: Configuring the Half-Opened Session Limit for an Inspect VRF Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type inspect vrf vrf1-pmap
Device(config-profile)# alert on
Device(config-profile)# max-incomplete 3500
Device(config-profile)# session total 34500
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
```

```
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# end
```

Example: Configuring the Global TCP SYN Flood Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box tcp syn-flood limit 500
Device(config-profile)# end
```

Additional References for Protection Against Distributed Denial of Service Attacks

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	Cisco IOS Security Command Reference
Firewall resource management	*Configuring Firewall Resource Management feature*
Firewall TCP SYN cookie	*Configuring Firewall TCP SYN Cookie feature*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Protection Against Distributed Denial of Service Attacks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 187: Feature Information for Protection Against Distributed Denial of Service Attacks

Feature Name	Releases	Feature Information
Protection Against Distributed Denial of Service Attacks	Cisco IOS XE Release 3.4S	The Protection Against Distributed Denial of Service Attacks feature provides protection from DoS attacks at the per-box level (for all firewall sessions) and at the VRF level. You can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, the half-opened connections limit, and global TCP SYN cookie protection to prevent DDoS attacks. The following commands were introduced or modified: **clear policy-firewall stats global**, **max-incomplete**, **max-incomplete aggressive-aging**, **per-box aggressive-aging**, **per-box max-incomplete**, **per-box max-incomplete aggressive-aging**, **per-box tcp syn-flood limit**, **session total**, **show policy-firewall stats global**, **show policy-firewall stats zone**, **threat-detection basic-threat**, **threat-detection rate**, and **udp half-open**.

CHAPTER 140

Configuring Firewall Resource Management

The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a router.

Restrictions for Configuring Firewall Resource Management

- After you configure the global-level or VRF-level session limit and reconfigure the session limit, if the global-level or VRF-level session limit is below the initially configured session count, no new session is added; however, no current session is dropped.

Information About Configuring Firewall Resource Management

Firewall Resource Management

Resource Management limits the level of usage of shared resources on a device. Shared resources on a device include:

- Bandwidth
- Connection states
- Memory usage (per table)
- Number of sessions or calls
- Packets per second
- Ternary content addressable memory (TCAM) entries

The Firewall Resource Management feature extends the zone-based firewall resource management from the class level to the VRF level and the global level. Class-level resource management provides resource protection for firewall sessions at a class level. For example, parameters such as the maximum session limit, the session rate limit, and the incomplete session limit protect firewall resources (for example, chunk memory) and keep these resources from being used up by a single class.

When virtual routing and forwarding (VRF) instances share the same policy, a firewall session setup request from one VRF instance can make the total session count reach the maximum limit. When one VRF consumes the maximum amount of resources on a device, it becomes difficult for other VRF instances to share device resources. To limit the number of VRF firewall sessions, you can use the Firewall Resource Management feature.

At the global level, the Firewall Resource Management feature helps limit the usage of resources at the global routing domain by firewall sessions.

VRF-Aware Cisco IOS XE Firewall

The VRF-Aware Cisco IOS XE Firewall applies the Cisco IOS XE Firewall functionality to VPN Routing and Forwarding (VRF) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge routers. SPs provide managed services to small and medium business markets.

The VRF-Aware Cisco IOS XE Firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

The VRF-aware firewall supports VRF-lite (also known as Multi-VRF CE) and Application Inspection and Control (AIC) for various protocols.

Note Cisco IOS XE Releases do not support Context-Based Access Control (CBAC) firewalls.

Firewall Sessions

Session Definition

At the virtual routing and forwarding (VRF) level, the Firewall Resource Management feature tracks the firewall session count for each VRF instance. At the global level, the firewall resource management tracks the total firewall session count at the global routing domain and not at the device level. In both the VRF and global levels, session count is the sum of opened sessions, half-opened sessions, and sessions in the imprecise firewall session database. A TCP session that has not yet reached the established state is called a half-opened session.

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on).

The following rules apply to the configuration of a session limit:

- The class-level session limit can exceed the global limit.
- The class-level session limit can exceed its associated VRF session maximum.
- The sum of the VRF limit, including the global context, can be greater than the hardcoded session limit.

Session Rate

The session rate is the rate at which sessions are established at any given time interval. You can define maximum and minimum session rate limits. When the session rate exceeds the maximum specified rate, the firewall starts rejecting new session setup requests.

From the resource management perspective, setting the maximum and minimum session rate limit helps protect Cisco Packet Processor from being overwhelmed when numerous firewall session setup requests are received.

Incomplete or Half-Opened Sessions

Incomplete sessions are half-opened sessions. Any resource used by an incomplete session is counted, and any growth in the number of incomplete sessions is limited by setting the maximum session limit.

Firewall Resource Management Sessions

The following rules apply to firewall resource management sessions:

- By default, the session limit for opened and half-opened sessions is unlimited.
- Opened or half-opened sessions are limited by parameters and counted separately.
- Opened or half-opened session count includes Internet Control Message Protocol (ICMP), TCP, or UDP sessions.
- You can limit the number and rate of opened sessions.
- You can only limit the number of half-opened sessions.

How to Configure Firewall Resource Management

Configuring Firewall Resource Management

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **session total** *number*
5. **tcp syn-flood limit** *number*
6. **exit**
7. **parameter-map type inspect-global**
8. **vrf** *vrf-name* **inspect** *parameter-map-name*
9. **exit**
10. **parameter-map type inspect-vrf vrf-default**

11. **session total** *number*
12. **tcp syn-flood limit** *number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 4	**session total** *number* **Example:** `Device(config-profile)# session total 1000`	Configures the total number of sessions.
Step 5	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 2000`	Limits the number of TCP half-opened sessions that trigger synchronization (SYN) cookie processing for new SYN packets.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect-global** **Example:** `Device(config)# parameter-map type inspect-global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 8	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF to the parameter map.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**parameter-map type inspect-vrf vrf-default** **Example:**	Configures a default inspect VRF-type parameter map.

	Command or Action	Purpose
	`Device(config)# parameter-map type inspect-vrf vrf-default`	
Step 11	**session total** *number* **Example:** `Device(config-profile)# session total 6000`	Configures the total number of sessions. • You can configure the **session total** command for an inspect VRF-type parameter map and for a global parameter map. When you configure the **session total** command for an inspect VRF-type parameter map, the sessions are associated with an inspect VRF-type parameter map. The **session total** command is applied to the global routing domain when it is configured for a global parameter-map.
Step 12	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 7000`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 13	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall Resource Management

Example: Configuring Firewall Resource Management

```
Device# configure terminal
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# session total 1000
Device(config-profile)# tcp syn-flood limit 2000
Device(config-profile)# exit
Device(config)# parameter-map type inspect-global
Device(config-profile)# vrf vrf1 inspect pmap1
Device(config-profile)# exit
Device(config)# parameter-map type inspect-vrf vrf-default
Device(config-profile)# session total 6000
Device(config-profile)# tcp syn-flood limit 7000
Device(config-profile)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
VRF-aware firewall	"VRF-Aware Cisco IOS XE Firewall" module
Zone-based policy firewall	" Zone-Based Policy Firewall" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Firewall Resource Management

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 188: Feature Information for Configuring Firewall Resource Management

Feature Name	Releases	Feature Information
Firewall Resource Management	Cisco IOS XE Release 3.3S	The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a router. The following commands were introduced or modified: **parameter-map type inspect-vrf**.

CHAPTER 141

IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

IPv6 zone-based firewalls support the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features.

The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. With the Protection Against Distributed Denial of Service Attacks feature, you can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, half-opened connections limit, and global TCP synchronization (SYN) cookie protection to prevent distributed DoS attacks.

The Firewall Resource Management feature limits the number of VPN Routing and Forwarding (VRF) and global firewall sessions that are configured on a device.

This module describes how to configure the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features.

Restrictions for IPv6 Firewall Support for Protection Against Distributed Denial of Service Attacks and Resource Management

The following restriction applies to the Firewall Resource Management feature:

- After you configure the global-level or the virtual routing and forwarding (VRF)-level session limit and reconfigure the session limit, if the global-level or the VRF-level session limit is below the initially configured session count, no new session is added; however, no current session is dropped.

Information About IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Aggressive Aging of Firewall Sessions

The Aggressive Aging feature provides the firewall the capability of aggressively aging out sessions to make room for new sessions, thereby protecting the firewall session database from filling. The firewall protects its resources by removing idle sessions. The Aggressive Aging feature allows firewall sessions to exist for a shorter period of time defined by a timer called aging-out time.

The Aggressive Aging feature includes thresholds to define the start and end of the aggressive aging period—high and low watermarks. The aggressive aging period starts when the session table crosses the high watermark and ends when it falls below the low watermark. During the aggressive aging period, sessions will exist for a shorter period of time that you have configured by using the aging-out time. If an attacker initiates sessions at a rate that is faster than the rate at which the firewall terminates sessions, all resources that are allocated for creating sessions are used and all new connections are rejected. To prevent such attacks, you can configure the Aggressive Aging feature to aggressively age out sessions. This feature is disabled by default.

You can configure aggressive aging for half-opened sessions and total sessions at the box level (box refers to the entire firewall session table) and the virtual routing and forwarding (VRF) level. If you have configured this feature for total sessions, all sessions that consume firewall session resources are taken into account. Total sessions comprise established sessions, half-opened sessions, and sessions in the imprecise session database. (A TCP session that has not yet reached the established state is called a half-opened session.)

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (the source IP address, the destination IP address, the source port, the destination port, and the protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on). In the case of aggressive aging for half-opened sessions, only half-opened sessions are considered.

You can configure an aggressive aging-out time for Internet Control Message Protocol (ICMP), TCP, and UDP firewall sessions. The aging-out time is set by default to the idle time.

Event Rate Monitoring Feature

The Event Rate Monitoring feature monitors the rate of predefined events in a zone. The Event Rate Monitoring feature includes basic threat detection, which is the ability of a security device to detect possible threats, anomalies, and attacks to resources inside the firewall and to take action against them. You can configure a basic threat detection rate for events. When the incoming rate of a certain type of event exceeds the configured threat detection rate, event rate monitoring considers this event as a threat and takes action to stop the threat. Threat detection inspects events only on the ingress zone (if the Event Rate Monitoring feature is enabled on the ingress zone).

The network administrator is informed about the potential threats via an alert message (syslog or high-speed logger [HSL]) and can take actions such as detecting the attack vector, detecting the zone from which the attack is coming, or configuring devices in the network to block certain behaviors or traffic.

The Event Rate Monitoring feature monitors the following types of events:

- Firewall drops due to basic firewall checks failure—This can include zone or zone-pair check failures, or firewall policies configured with the drop action, and so on.
- Firewall drops due to Layer 4 inspection failure—This can include TCP inspections that have failed because the first TCP packet is not a synchronization (SYN) packet.
- TCP SYN cookie attack—This can include counting the number of SYN packets that are dropped and the number of SYN cookies that are sent as a spoofing attack.

The Event Rate Monitoring feature monitors the average rate and the burst rate of different events. Each event type has a rate object that is controlled by an associated rate that has a configurable parameter set (the average threshold, the burst threshold, and a time period). The time period is divided into time slots; each time slot is 1/30th of the time period.

The average rate is calculated for every event type. Each rate object holds 30 completed sampling values plus one value to hold the current ongoing sampling period. The current sampling value replaces the oldest calculated value and the average is recalculated. The average rate is calculated during every time period. If the average rate exceeds the average threshold, the Event Rate Monitoring feature will consider this as a possible threat, update the statistics, and inform the network administrator.

The burst rate is implemented by using the token bucket algorithm. For each time slot, the token bucket is filled with tokens. For each event that occurs (of a specific event type), a token is removed from the bucket. An empty bucket means that the burst threshold is reached, and the administrator receives an alarm through the syslog or HSL. You can view the threat detection statistics and learn about possible threats to various events in the zone from the output of the **show policy-firewall stats zone** command.

You must first enable basic threat detection by using the **threat-detection basic-threat** command. Once basic threat detection is configured, you can configure the threat detection rate. To configure the threat detection rate, use the **threat-detection rate** command.

The following table describes the basic threat detection default settings that are applicable if the Event Rate Monitoring feature is enabled.

Table 189: Basic Threat Detection Default Settings

Packet Drop Reason	Threat Detection Settings
Basic firewall drops	average-rate 400 packets per second (pps) burst-rate 1600 pps rate-interval 600 seconds
Inspection-based firewall drops	average-rate 400 pps burst-rate 1600 pps rate-interval 600 seconds
SYN attack firewall drops	average-rate 100 pps burst-rate 200 pps rate-interval 600 seconds

Half-Opened Connections Limit

The firewall session table supports the limiting of half-opened firewall connections. Limiting the number of half-opened sessions will defend the firewall against attacks that might fill the firewall session table at the per-box level or at the virtual routing and forwarding (VRF) level with half-opened sessions and prevent sessions from being established. The half-opened connection limit can be configured for Layer 4 protocols, Internet Control Message Protocol (ICMP), TCP, and UDP. The limit set to the number of UDP half-opened sessions will not affect the TCP or ICMP half-opened sessions. When the configured half-opened session limit is exceeded, all new sessions are rejected and a log message is generated, either in syslog or in the high-speed logger (HSL).

The following sessions are considered as half-opened sessions:

- TCP sessions that have not completed the three-way handshake.
- UDP sessions that have only one packet detected in the UDP flow.
- ICMP sessions that do not receive a reply to the ICMP echo request or the ICMP time-stamp request.

TCP SYN-Flood Attacks

You can configure the global TCP SYN-flood limit to limit SYN flood attacks. TCP SYN-flooding attacks are a type of denial of service (DoS) attack. When the configured TCP SYN-flood limit is reached, the firewall verifies the source of sessions before creating more sessions. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or program tries to use false data to gain access to resources in a network. TCP SYN flooding can take up all resources on a firewall or an end host, thereby causing denial of service to legitimate traffic. You can configure TCP SYN-flood protection at the VRF level and the zone level.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host intending to utilize all resources on that host.

- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the intention of exhausting the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall.

Firewall Resource Management

Resource Management limits the level of usage of shared resources on a device. Shared resources on a device include:

- Bandwidth
- Connection states
- Memory usage (per table)
- Number of sessions or calls
- Packets per second
- Ternary content addressable memory (TCAM) entries

The Firewall Resource Management feature extends the zone-based firewall resource management from the class level to the VRF level and the global level. Class-level resource management provides resource protection for firewall sessions at a class level. For example, parameters such as the maximum session limit, the session rate limit, and the incomplete session limit protect firewall resources (for example, chunk memory) and keep these resources from being used up by a single class.

When virtual routing and forwarding (VRF) instances share the same policy, a firewall session setup request from one VRF instance can make the total session count reach the maximum limit. When one VRF consumes the maximum amount of resources on a device, it becomes difficult for other VRF instances to share device resources. To limit the number of VRF firewall sessions, you can use the Firewall Resource Management feature.

At the global level, the Firewall Resource Management feature helps limit the usage of resources at the global routing domain by firewall sessions.

Firewall Sessions

Session Definition

At the virtual routing and forwarding (VRF) level, the Firewall Resource Management feature tracks the firewall session count for each VRF instance. At the global level, the firewall resource management tracks the total firewall session count at the global routing domain and not at the device level. In both the VRF and global levels, session count is the sum of opened sessions, half-opened sessions, and sessions in the imprecise firewall session database. A TCP session that has not yet reached the established state is called a half-opened session.

A firewall has two session databases: the session database and the imprecise session database. The session database contains sessions with 5-tuple (source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. The imprecise session database contains sessions with fewer than 5-tuple (missing IP addresses, port numbers, and so on).

The following rules apply to the configuration of a session limit:

- The class-level session limit can exceed the global limit.
- The class-level session limit can exceed its associated VRF session maximum.
- The sum of the VRF limit, including the global context, can be greater than the hardcoded session limit.

Session Rate

The session rate is the rate at which sessions are established at any given time interval. You can define maximum and minimum session rate limits. When the session rate exceeds the maximum specified rate, the firewall starts rejecting new session setup requests.

From the resource management perspective, setting the maximum and minimum session rate limit helps protect Cisco Packet Processor from being overwhelmed when numerous firewall session setup requests are received.

Incomplete or Half-Opened Sessions

Incomplete sessions are half-opened sessions. Any resource used by an incomplete session is counted, and any growth in the number of incomplete sessions is limited by setting the maximum session limit.

Firewall Resource Management Sessions

The following rules apply to firewall resource management sessions:

- By default, the session limit for opened and half-opened sessions is unlimited.
- Opened or half-opened sessions are limited by parameters and counted separately.
- Opened or half-opened session count includes Internet Control Message Protocol (ICMP), TCP, or UDP sessions.
- You can limit the number and rate of opened sessions.
- You can only limit the number of half-opened sessions.

How to Configure IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf-definition** *vrf-name* **Example:** `Device(config)# vrf-definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes.
Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect ipv6-param-map`	Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
Step 8	**sessions maximum** *sessions* **Example:** `Device(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 11	**ip port-map** *appl-name* **port** *port-num* **list** *list-name* **Example:** `Device(config)# ip port-map ftp port 8090 list ipv6-acl`	Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL).
Step 12	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 13	**permit ipv6 any any** **Example:** `Device(config-ipv6-acl)# permit ipv6 any any`	Sets permit conditions for an IPv6 access list.
Step 14	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 15	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ipv6-class`	Creates an application-specific inspect type class map and enters QoS class-map configuration mode.
Step 16	**match access-group name** *access-group-name* **Example:**	Configures the match criteria for a class map on the basis of the specified ACL.

	Command or Action	Purpose
	`Device(config-cmap)# match access-group name ipv6-acl`	
Step 17	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criterion for a class map on the basis of the specified protocol.
Step 18	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 19	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ipv6-policy`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 20	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ipv6-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 21	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect ipv6-param-map`	Enables stateful packet inspection.
Step 22	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring the Aggressive Aging of Firewall Sessions

You can configure the Aggressive Aging feature for per-box (per-box refers to the entire firewall session table), default-VRF, and per-VRF firewall sessions. Before the Aggressive Aging feature can work, you must configure the aggressive aging and the aging-out time of firewall sessions.

Perform the following tasks to configure the aggressive aging of firewall sessions.

Configuring per-Box Aggressive Aging

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**

- **parameter-map type inspect global**

4. **per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
5. **per-box aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Steps 4 and 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**per-box max-incomplete** *number* **aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# per-box max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging rate for half-opened sessions in the firewall session table.
Step 5	**per-box aggressive-aging high** {*value* **low** *value*	**percent** *percent* **low percent** *percent*}	Configures the aggressive aging limit of total sessions.

	Command or Action	Purpose
	Example: `Device(config-profile)# per-box aggressive-aging high 1700 low 1300`	
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring Aggressive Aging for a Default VRF

When you configure the **max-incomplete aggressive-aging** `command, it applies to the default VRF.`

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enters one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}

5. **session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}]
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **end**
10. **show policy-firewall stats vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enters one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255`	Configures the maximum limit and the aggressive aging limit of half-opened firewall sessions.
Step 5	**session total** *number* [**aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total limit and the aggressive aging limit for total firewall sessions.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • After aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 10	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	Displays global VRF firewall policy statistics.

Configuring per-VRF Aggressive Aging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **parameter-map type inspect-vrf** *vrf-pmap-name*
9. **max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*}
10. **session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}]
11. **alert on**
12. **exit**

13. Enter one of the following commands:
    - **parameter-map type inspect-global**
    - **parameter-map type inspect global**
14. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
15. **exit**
16. **parameter-map type inspect** *parameter-map-name*
17. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
18. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
19. **exit**
20. **policy-map type inspect** *policy-map-name*
21. **class type inspect match-any** *class-map-name*
22. **inspect** *parameter-map-name*
23. **end**
24. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf ddos-vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:2`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:2`	Creates a route-target extended community and exports the routing information to the target VPN extended community.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:2`	Creates a route-target extended community and imports routing information from the target VPN extended community.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 8	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 9	**max-incomplete** *number* **aggressive-aging high** {*value* **low** *value* | **percent** *percent* **low percent** *percent*} **Example:** `Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200`	Configures the maximum limit and the aggressive aging limit for half-opened sessions.
Step 10	**session total** *number* [**aggressive-aging** {**high** *value* **low** *value* | **percent** *percent* **low percent** *percent*}] **Example:** `Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60`	Configures the total session limit and the aggressive aging limit for the total sessions. • You can configure the total session limit as an absolute value or as a percentage.
Step 11	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 12	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 13	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 14 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 14	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.

	Command or Action	Purpose
Step 15	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 16	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 17	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions.
Step 18	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is disabled when the connections drop below the low watermark.
Step 19	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 20	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 21	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 22	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 23	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 24	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
       Total Session Count(estab + half-open): 80, Exceed: 0
       Total Session Aggressive Aging Period Off, Event Count: 0

                Half Open
       Protocol Session Cnt     Exceed
       -------- -----------     ------
       All      0               0
       UDP      0               0
       ICMP     0               0
       TCP      0               0

       TCP Syn Flood Half Open Count: 0, Exceed: 116
       Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring the Aging Out of Firewall Sessions

You can configure the aging out of ICMP, TCP, or UDP firewall sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
5. **exit**
6. **parameter-map type inspect** *parameter-map-name*
7. **tcp idle-time** *seconds* [**ageout-time** *seconds*]
8. **tcp synwait-time** *seconds* [**ageout-time** *seconds*]
9. **exit**
10. **policy-map type inspect** *policy-map-name*
11. **class type inspect match-any** *class-map-name*
12. **inspect** *parameter-map-name*
13. **end**
14. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspectglobal`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip Step 4 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF with a parameter map.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap1`	Configures an inspect-type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action and enters parameter-map type inspect configuration mode.
Step 7	**tcp idle-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp idle-time 3000 ageout-time 100`	Configures the timeout for idle TCP sessions and the aggressive aging-out time for TCP sessions. • You can also configure the **tcp finwait-time** command to specify how long a TCP session will be managed after the firewall detects a finish (FIN) exchange, or you can configure the **tcp synwait-time** command to specify how long the software will wait

	Command or Action	Purpose
		for a TCP session to reach the established state before dropping the session.
Step 8	**tcp synwait-time** *seconds* [**ageout-time** *seconds*] **Example:** `Device(config-profile)# tcp synwait-time 30 ageout-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session. • When aggressive aging is enabled, the SYN wait timer of the oldest TCP connections are reset from the default to the configured ageout time. In this example, instead of waiting for 30 seconds for connections to timeout, the timeout of the oldest TCP connections are set to 10 seconds. Aggressive aging is enabled when the connections drop below the low watermark.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ddos-fw`	Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode.
Step 11	**class type inspect match-any** *class-map-name* **Example:** `Device(config-pmap)# class type inspect match-any ddos-class`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 12	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap1`	Enables stateful packet inspection for the parameter map.
Step 13	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.
Step 14	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Example

The following is sample output from the **show policy-firewall stats vrf vrf1-pmap** command:

```
Device# show policy-firewall stats vrf vrf1-pmap

 VRF: vrf1, Parameter-Map: vrf1-pmap
  Interface reference count: 2
```

```
Total Session Count(estab + half-open): 270, Exceed: 0
Total Session Aggressive Aging Period Off, Event Count: 0

         Half Open
Protocol Session Cnt     Exceed
-------- -----------     ------
All      0               0
UDP      0               0
ICMP     0               0
TCP      0               0

TCP Syn Flood Half Open Count: 0, Exceed: 12
Half Open Aggressive Aging Period Off, Event Count: 0
```

Configuring Firewall Event Rate Monitoring

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **alert on**
5. **threat-detection basic-threat**
6. **threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
7. **threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
8. **threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second*
9. **exit**
10. **zone security** *security-zone-name*
11. **protection** *parameter-map-name*
12. **exit**
13. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
14. **end**
15. **show policy-firewall stats zone**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**parameter-map type inspect-zone** *zone-pmap-name* **Example:** `Device(config)# parameter-map type inspect-zone zone-pmap1`	Configures an inspect-zone parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages for a zone. • You can use the **log** command to configure the logging of alerts either to the syslog or to the high-speed logger (HSL).
Step 5	**threat-detection basic-threat** **Example:** `Device(config-profile)# threat-detection basic-threat`	Configures basic threat detection for a zone.
Step 6	**threat-detection rate fw-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall drop events. • You must configure the **threat-detection basic-threat** command before you configure the **threat-detection rate** command.
Step 7	**threat-detection rate inspect-drop average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for firewall inspection-based drop events.
Step 8	**threat-detection rate syn-attack average-time-frame** *seconds* **average-threshold** *packets-per-second* **burst-threshold** *packets-per-second* **Example:** `Device(config-profile)# threat-detection rate syn-attack average-time-frame 600 average-threshold 100 burst-threshold 100`	Configures the threat detection rate for TCP SYN attack events.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**zone security** *security-zone-name* **Example:** `Device(config)# zone security public`	Creates a security zone and enters security zone configuration mode.

	Command or Action	Purpose
Step 11	**protection** *parameter-map-name* **Example:** `Device(config-sec-zone)# protection zone-pmap1`	Attaches the inspect-zone parameter map to the zone and applies the features configured in the inspect-zone parameter map to the zone.
Step 12	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 13	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private2public source private destination public`	Creates a zone pair and enters security zone-pair configuration mode.
Step 14	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and enters privileged EXEC mode.
Step 15	**show policy-firewall stats zone** **Example:** `Device# show policy-firewall stats zone`	Displays policy firewall statistics at the zone level.

Configuring the per-Box Half-Opened Session Limit

Per-box refers to the entire firewall session table. Any configuration that follows the **parameter-map type inspect-global** command applies to the box.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
4. **alert on**
5. **per-box max-incomplete** *number*
6. **session total** *number*
7. **end**
8. **show policy-firewall stats global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, the **parameter-map type inspect-global** and the **parameter-map type inspect global** commands are supported. You cannot configure both these commands together.
		• Skip to Steps 5 and 6 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box max-incomplete** *number* **Example:** `Device(config-profile)# per-box max-incomplete 12345`	Configures the maximum number of half-opened connections for the firewall session table.
Step 6	**session total** *number* **Example:** `Device(config-profile)# session total 34500`	Configures the total session limit for the firewall session table.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 8	**show policy-firewall stats global** **Example:** `Device# show policy-firewall stats global`	Displays global firewall statistics information.

Configuring the Half-Opened Session Limit for an Inspect-VRF Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-name*
4. **alert on**
5. **max-incomplete** *number*
6. **session total** *number*
7. **exit**
8. Enter one of the following commands:
   - **parameter-map type inspect-global**
   - **parameter-map type inspect global**
9. **alert on**
10. **vrf** *vrf-name* **inspect** *vrf-pmap-name*
11. **end**
12. **show policy-firewall stats vrf** *vrf-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect-VRF parameter map and enters parameter-map type inspect configuration mode.
Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**max-incomplete** *number* **Example:** `Device(config-profile)# max-incomplete 2000`	Configures the maximum number of half-opened connections per VRF.
Step 6	**session total** *number* **Example:**	Configures the total session limit for a VRF.

	Command or Action	Purpose
	`Device(config-profile)# session total 34500`	
Step 7	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.

Step 8	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map for connecting thresholds and timeouts and enters parameter-map type inspect configuration mode.
		• Based on your release, you can use either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 10 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 9	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 10	**vrf** *vrf-name* **inspect** *vrf-pmap-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds the VRF to the global parameter map.
Step 11	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.
Step 12	**show policy-firewall stats vrf** *vrf-pmap-name* **Example:** `Device# show policy-firewall stats vrf vrf1-pmap`	Displays VRF-level policy firewall statistics.

Configuring the Global TCP SYN Flood Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:

- **parameter-map type inspect-global**
- **parameter-map type inspect global**

4. **alert on**
5. **per-box tcp syn-flood limit** *number*
6. **end**
7. **show policy-firewall stats vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **parameter-map type inspect-global**
	• **parameter-map type inspect global**
	Example:
	`Device(config)# parameter-map type inspect-global`
	`Device(config)# parameter-map type inspect global`
		Configures a global parameter map and enters parameter-map type inspect configuration mode.
		• Based on your release, you can configure either the **parameter-map type inspect-global** command or the **parameter-map type inspect global** command. You cannot configure both these commands together.
		• Skip Step 5 if you configure the **parameter-map type inspect-global** command.
		Note If you configure the **parameter-map type inspect-global** command, **per-box** configurations are not supported because, by default, all **per-box** configurations apply to all firewall sessions.

Step 4	**alert on** **Example:** `Device(config-profile)# alert on`	Enables the console display of stateful packet inspection alert messages.
Step 5	**per-box tcp syn-flood limit** *number* **Example:** `Device(config-profile)# per-box tcp syn-flood limit 500`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show policy-firewall stats vrf global** **Example:** `Device# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-opened sessions are present.

Example

The following is sample output from the **show policy-firewall stats vrf global** command:

```
Device# show policy-firewall stats vrf global

Global table statistics
       total_session_cnt: 0
       exceed_cnt:        0
       tcp_half_open_cnt: 0
       syn_exceed_cnt:    0
```

Configuring Firewall Resource Management

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **session total** *number*
5. **tcp syn-flood limit** *number*
6. **exit**
7. **parameter-map type inspect-global**
8. **vrf** *vrf-name* **inspect** *parameter-map-name*
9. **exit**
10. **parameter-map type inspect-vrf vrf-default**
11. **session total** *number*
12. **tcp syn-flood limit** *number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Device(config)# parameter-map type inspect-vrf vrf1-pmap`	Configures an inspect VRF-type parameter map and enters parameter-map type inspect configuration mode.
Step 4	**session total** *number* **Example:** `Device(config-profile)# session total 1000`	Configures the total number of sessions.
Step 5	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 2000`	Limits the number of TCP half-opened sessions that trigger synchronization (SYN) cookie processing for new SYN packets.
Step 6	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 7	**parameter-map type inspect-global** **Example:** `Device(config)# parameter-map type inspect-global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 8	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Device(config-profile)# vrf vrf1 inspect vrf1-pmap`	Binds a VRF to the parameter map.
Step 9	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and enters global configuration mode.
Step 10	**parameter-map type inspect-vrf vrf-default** **Example:** `Device(config)# parameter-map type inspect-vrf vrf-default`	Configures a default inspect VRF-type parameter map.
Step 11	**session total** *number* **Example:** `Device(config-profile)# session total 6000`	Configures the total number of sessions. • You can configure the **session total** command for an inspect VRF-type parameter map and for a global parameter map. When you configure the **session total** command for an inspect VRF-type parameter map, the sessions are associated with an inspect VRF-type

	Command or Action	Purpose
		parameter map. The **session total** command is applied to the global routing domain when it is configured for a global parameter-map.
Step 12	**tcp syn-flood limit** *number* **Example:** `Device(config-profile)# tcp syn-flood limit 7000`	Limits the number of TCP half-opened sessions that trigger SYN cookie processing for new SYN packets.
Step 13	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

Example: Configuring the Aggressive Aging of Firewall Sessions

Example: Configuring per-Box Aggressive Aging

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# per-box max-incomplete 2000 aggressive-aging 1500 low 1200
Device(config-profile)# per-box aggressive-aging high 1700 low 1300
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring Aggressive Aging for a Default VRF

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# max-incomplete 2000 aggressive-aging high 1500 low 1200
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# end
```

Example: Configuring per-VRF Aggressive Aging

```
Device# configure terminal
Device(config)# ip vrf ddos-vrf1
Device(config-vrf)# rd 100:2
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# exit
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# max-incomplete 3455 aggressive-aging high 2345 low 2255
Device(config-profile)# session total 1000 aggressive-aging high percent 80 low percent 60
Device(config-profile)# alert on
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-pmap)# class type inspect match-any ddos-class
Device(config-pmap-c)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring the Aging Out of Firewall Sessions

```
Device# configure terminal
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
```

```
Device(config-profile)# exit
Device(config)# parameter-map type inspect pmap1
Device(config-profile)# tcp idle-time 3000 ageout-time 100
Device(config-profile)# tcp synwait-time 30 ageout-time 10
Device(config-profile)# exit
Device(config)# policy-map type inspect ddos-fw
Device(config-profile)# class type inspect match-any ddos-class
Device(config-profile)# inspect pmap1
Device(config-profile)# end
```

Example: Configuring Firewall Event Rate Monitoring

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect zone zone-pmap1
Device(config-profile)# alert on
Device(config-profile)# threat-detection basic-threat
Device(config-profile)# threat-detection rate fw-drop average-time-frame 600 average-threshold
 100 burst-threshold 100
Device(config-profile)# threat-detection rate inspect-drop average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# threat-detection rate syn-attack average-time-frame 600
average-threshold 100 burst-threshold 100
Device(config-profile)# exit
Device(config)# zone security public
Device(config-sec-zone)# protection zone-pmap1
Device(config-sec-zone)# exit
Device(config)# zone-pair security private2public source private destination public
Device(config-sec-zone-pair)# end
```

Example: Configuring the per-Box Half-Opened Session Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box max-incomplete 12345
Device(config-profile)# session total 34500
Device(config-profile)# end
```

Example: Configuring the Half-Opened Session Limit for an Inspect VRF Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type inspect vrf vrf1-pmap
Device(config-profile)# alert on
Device(config-profile)# max-incomplete 3500
Device(config-profile)# session total 34500
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
```

```
Device(config-profile)# vrf vrf1 inspect vrf1-pmap
Device(config-profile)# end
```

Example: Configuring the Global TCP SYN Flood Limit

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# alert on
Device(config-profile)# per-box tcp syn-flood limit 500
Device(config-profile)# end
```

Example: Configuring Firewall Resource Management

```
Device# configure terminal
Device(config)# parameter-map type inspect-vrf vrf1-pmap
Device(config-profile)# session total 1000
Device(config-profile)# tcp syn-flood limit 2000
Device(config-profile)# exit
Device(config)# parameter-map type inspect-global
Device(config-profile)# vrf vrf1 inspect pmap1
Device(config-profile)# exit
Device(config)# parameter-map type inspect-vrf vrf-default
Device(config-profile)# session total 6000
Device(config-profile)# tcp syn-flood limit 7000
Device(config-profile)# end
```

Additional References for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 190: Feature Information for IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management

Feature Name	Releases	Feature Information
IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management	Cisco IOS XE Release 3.7S	IPv6 zone-based firewalls support the Protection of Distributed Denial of Service Attacks and the Firewall Resource Management features. The Protection Against Distributed Denial of Service Attacks feature provides protection from Denial of Service (DoS) attacks at the global level (for all firewall sessions) and at the VPN routing and forwarding (VRF) level. You can configure the aggressive aging of firewall sessions, event rate monitoring of firewall sessions, half-opened connections limit, and global TCP SYN cookie protection to prevent distributed DoS attacks. The Firewall Resource Management feature limits the number of VPN routing and forwarding (VRF) instances and global firewall sessions that are configured on a device.
IPv6 Firewall Support for Prevention of Distributed Denial of Service Attacks and Resource Management	Cisco IOS XE Release 3.10S	In Cisco IOS XE Release 3.10S, support was added for Cisco CSR 1000V Series Routers.

CHAPTER 142

Configurable Number of Simultaneous Packets per Flow

In zone-based policy firewalls, the number of simultaneous packets per flow is restricted to 25 and packets that exceed the limit are dropped. The dropping of packets when the limit is reached impacts the performance of networks. The Configurable Number of Simultaneous Packets per Flow feature allows you to configure the number of simultaneous packets per flow from 25 to 100.

This modules provides an overview of the feature and explains how to configure it.

Restrictions for Configurable Number of Simultaneous Packets per Flow

- When the TCP window scale option is configured, the firewall cannot simultaneously fit too many TCP packets per flow, and packets that exceed the configured limit are dropped. The maximum window size that can be used, if the TCP window scale option is enabled, is 1 GB.

  The standard TCP window size is between 2 and 65,535 bytes. If the TCP payload size is smaller than 655 bytes, 100 simultaneous packets cannot contain all TCP packets that belong to a single TCP window, and this can result in packet drops. We recommend that you increase the TCP payload size or reduce the TCP window size to avoid packet drops.

- The total available threads in each platform varies according to the enabled license levels. If the configured number of simultaneous packets per flow is bigger than the available hardware thread number, the configuration of simultaneous packets is not effective.

Information About Configurable Number of Simultaneous Packets per Flow

Overview of Configurable Number of Simultaneous Packets per Flow

The Configurable Number of Simultaneous Packets per Flow feature allows you to increase the number of simultaneous packets per flow that can enter a network. You can increase the number of simultaneous packets per flow from 25 to 100. The default is 25 simultaneous packets.

In multithreaded environments, the zone-based policy firewall may simultaneously receive multiple packets for a single traffic flow. During packet processing, the firewall uses two types of locks: flow lock and software lock. The flow lock ensures that packets that belong to the same flow are processed in the correct order. Normal software locks are used when multiple power processing element (PPE) threads try to read or write critical sections or common data structure (for example, memory).

If the number of simultaneous packets per flow is too large, the time taken by a thread to request and acquire a lock may be too long. This latency adversely affects time-critical infrastructure such as resource reuse and heat-beat processing. To control latency, the number of simultaneous packets was restricted to 25, and packets that exceeded 25 were dropped.

However, the dropping of packets drastically impacts system performance of a system. To minimize packet dropping, the Configurable Number of Simultaneous Packets per Flow feature was introduced. You can configure the number of simultaneous packets per flow from 25 to 100.

To change the number of simultaneous packets per flow, you must configure either the **parameter-map type inspect** *parameter-map-name* command or the **parameter-map type inspect global** command, followed by the **session packet** command. The limit configured under the **parameter-map type inspect** *parameter-map-name* command takes precedence over the limit configured under the **parameter-map type inspect global** command.

The firewall considers Session Initiation Protocol (SIP) trunk traffic as a single session. However, the SIP trunk traffic contains a large number of application-layer gateway (ALG) flows of different users. When the throughput of the SIP trunk traffic is high compared to other traffic, the simultaneous packet limit causes packets to drop and users may experience call drops.

How to Configure the Number of Simultaneous Packets per Flow

Configuring Class Maps and Policy Maps for Simultaneous Packets per Flow

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol** *protocol-name*

5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode. • Enter your password if prompted.
Step 3	**class-map type inspect** {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any cmap-protocols`	Creates an inspect-type class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol tcp`	Configures the match criteria for a class map on the basis of a specified protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class map configuration mode and returns to global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect policy1`	Creates an inspect-type policy map and enters policy map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect cmap-protocols`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Configures or modifies a policy for the default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy map configuration mode and returns to privileged EXEC mode.

Configuring the Number of Simultaneous Packets per Flow

You can configure the number of simultaneous packets per flow after configuring either the **parameter-map type inspect** command or the **parameter-map type inspect global** command. The number of simultaneous packets per flow configured under the **parameter-map type inspect** command overwrites the number configured under the **parameter-map type inspect global** command.

You must configure the **session packet** command to configure the number of simultaneous packets per flow.

Note You must configure either Steps 3 and 4 or Steps 6 and 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **session packet** *number-of-simultaneous-packets*
5. **exit**
6. **parameter-map type inspect global**
7. **session packet** *number-of-simultaneous-packets*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:**	Enters global configuration mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect param1`	(Optional) Defines an inspect type parameter map, which configures connection thresholds, timeouts, and other parameters pertaining to the inspect action; and enters parameter-map type inspect configuration mode.
Step 4	**session packet** *number-of-simultaneous-packets* **Example:** `Device(config-profile)# session packet 55`	(Optional) Configures the number of simultaneous traffic packets that can be configured per session. • Valid values for the *number-of-simultaneous-packets* argument are 25 to 55.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	(Optional) Defines a global inspect parameter map and enters parameter-map type inspect configuration mode.
Step 7	**session packet** *number-of-simultaneous-packets* **Example:** `Device(config-profile)# session packet 35`	(Optional) Configures the number of simultaneous traffic packets that can be configured per session. • Valid values for the *number-of-simultaneous-packets* argument are 25 to 55.
Step 8	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Configuring Zones for Simultaneous Packets per Flow

This task shows how to configure security zones, a zone pair, and assign interfaces as zone members.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone*
4. **exit**
5. **zone security** *security-zone*
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*

12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone* **Example:** `Device(config)# zone security z1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** *security-zone* **Example:** `Device(config)# zone security z2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security zp-security source z1 destination z2`	Creates a zone pair and enters security zone pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect policy1`	Attaches a firewall policy map to the destination zone pair. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:**	Exits security zone pair configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-sec-zone-pair)# exit`	
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z1`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone a part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/3`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Configurable Number of Simultaneous Packets per Flow

Example: Configuring Class Maps and Policy Maps for Simultaneous Packets per Flow

```
Device# configure terminal
Device(config)# class-map type inspect match-any cmap-protocols
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect policy1
Device(config-pmap)# class type inspect cmap-protocols
```

```
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# end
```

Example: Configuring the Number of Simultaneous Packets per Flow

You can configure the number of simultaneous packets per flow after configuring either the **parameter-map type inspect** command or the **parameter-map type inspect global** command. The number of simultaneous packets per flow configured under the **parameter-map type inspect** command overwrites the number configured under the **parameter-map type inspect global** command.

```
Device# configure terminal
Device(config)# parameter-map type inspect param1
Device(config-profile)# session packet 55
Device(config-profile)# exit
Device(config)# parameter-map type inspect global
Device(config-profile)# session packet 35
Device(config-profile)# end
```

Example: Configuring Zones for Simultaneous Packets per Flow

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security zp-security source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect policy1
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# zone-member security z1
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/3
Device(config-if)# zone-member security z2
Device(config-if)# end
```

Additional References for Configurable Number of Simultaneous Packets per Flow

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Configurable Number of Simultaneous Packets per Flow

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 191: Feature Information for Configurable Number of Simultaneous Packets per Flow

Feature Name	Releases	Feature Information
Configurable Number of Simultaneous Packets per Flow	Cisco IOS XE Release 3.11S	In zone-based policy firewalls, the number of simultaneous packets per flow was restricted to 25, and packets that exceeded the limit were dropped. The dropping of packets when the number is reached impacts network performance. The Configurable Number of Simultaneous Packets per Flow feature allows you to configure the number of simultaneous packets per flow from 25 to 100. In Cisco IOS XE Release 3.11S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers, the Cisco 4400 Series Integrated Services Routers, and the Cisco Cloud Services Routers 1000V Series. The following commands were introduced or modified: **session packet**, **show parameter-map type inspect**, **show platform hardware qfp feature firewall datapath scb**, **show platform hardware qfp feature firewall zone-pair**, and **show platform software firewall parameter-map**.

CHAPTER 143

Firewall High-Speed Logging

The Firewall High-Speed Logging feature supports the high-speed logging (HSL) of firewall messages by using NetFlow Version 9 as the export format.

This module describes how to configure HSL for zone-based policy firewalls.

Feature Information for Firewall High-Speed Logging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 192: Feature Information for Firewall High-Speed Logging

Feature Name	Releases	Feature Information
Firewall High-Speed Logging	Cisco IOS XE Release 2.1	The Firewall High-Speed Logging Support feature introduces support for the firewall HSL using NetFlow Version 9 as the export format. The following commands were introduced or modified: **log dropped-packet**, **log flow-export v9 udp destination**, **log flow-export template timeout-rate**, **parameter-map type inspect global**.
Configuring Zone-based Firewall using High-Speed Logging	Cisco IOS XE Gibraltar 16.11.1	In this release, support was added for the source interface. The following commands were introduced or modified: **log flow-export v9 udp destination source interface** *interface-name*

Information About Firewall High-Speed Logging

Firewall High-Speed Logging Overview

Zone-based firewalls support high-speed logging (HSL). When HSL is configured, a firewall provides a log of packets that flow through routing devices (similar to the NetFlow Version 9 records) to an external collector. Records are sent when sessions are created and destroyed. Session records contain the full 5-tuple information (the source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements.

HSL allows a firewall to log records with minimum impact to packet processing. The firewall uses buffered mode for HSL. In buffered mode, a firewall logs records directly to the high-speed logger buffer, and exports of packets separately.

Note High-Speed Logging (HSL) cannot be routed over VASI interfaces.

Note You can configure a maximum of 4 HSL destinations in a Zone-based firewall.

A firewall logs the following types of events:

- Audit—Session creation and removal notifications.
- Alert—Half-open and maximum-open TCP session notifications.
- Drop—Packet-drop notifications.
- Pass—Packet-pass (based on the configured rate limit) notifications.
- Summary—Policy-drop and pass-summary notifications.

The NetFlow collector issues the **show platform software interface F0 brief** command to map the FW_SRC_INTF_ID and FW_DST_INTF_ID interface IDs to the interface name.

The following sample output from the **show platform software interface F0 brief** command shows that the ID column maps the interface ID to the interface name (Name column):

```
Device# show platform software interface F0 brief

Name                            ID      QFP ID
GigabitEthernet0/2/0            16         9
GigabitEthernet0/2/1            17        10
GigabitEthernet0/2/2            18        11
GigabitEthernet0/2/3            19        12
```

NetFlow Field ID Descriptions

The following table lists NetFlow field IDs used within the firewall NetFlow templates:

Table 193: NetFlow Field IDs

Field ID	Type	Length	Description
NetFlow ID Fields (Layer 3 IPv4)			
FW_SRC_ADDR_IPV4	8	4	Source IPv4 address
FW_DST_ADDR_IPV4	12	4	Destination IPv4 address
FW_SRC_ADDR_IPV6	27	16	Source IPv6 address
FW_DST_ADDR_IPV6	28	16	Destination IPv6 address
FW_PROTOCOL	4	1	IP protocol value
FW_IPV4_IDENT	54	4	IPv4 identification
FW_IP_PROTOCOL_VERSION	60	1	IP protocol version
Flow ID Fields (Layer 4)			
FW_TCP_FLAGS	6	1	TCP flags
FW_SRC_PORT	7	2	Source port
FW_DST_PORT	11	2	Destination port
FW_ICMP_TYPE	176	1	ICMP [7] type value
FW_ICMP_CODE	177	1	ICMP code value
FW_ICMP_IPV6_TYPE	178	1	ICMP Version 6 (ICMPv6) type value
FW_ICMP_IPV6_CODE	179	1	ICMPv6 code value
FW_TCP_SEQ	184	4	TCP sequence number
FW_TCP_ACK	185	4	TCP acknowledgment number
Flow ID Fields (Layer 7)			
FW_L7_PROTOCOL_ID	95	2	Layer 7 protocol ID. Identifies the Layer 7 application classification used by firewall inspection. Normal records use 2 bytes, but optional records use 4 bytes.
Flow Name Fields (Layer 7)			
FLOW_FIELD_L7_PROTOCOL_NAME	96	32	Layer 7 protocol name. Identifies the Layer 7 protocol name that corresponds to the Layer 7 protocol ID (FW_L7_PROTOCOL_ID).
Flow ID Fields (Interface)			

Field ID	Type	Length	Description
FW_SRC_INTF_ID	10	2	Ingress SNMP [8] ifIndex
FW_DST_INTF_ID	14	2	Egress SNMP ifIndex
FW_SRC_VRF_ID	234	4	Ingress (initiator) VRF [9] ID
FW_DST_VRF_ID	235	4	Egress (responder) VRF ID
FW_VRF_NAME	236	32	VRF name
Mapped Flow ID Fields (Network Address Translation)			
FW_XLATE_SRC_ADDR_IPV4	225	4	Mapped source IPv4 address
FW_XLATE_DST_ADDR_IPV4	226	4	Mapped destination IPv4 address
FW_XLATE_SRC_PORT	227	2	Mapped source port
FW_XLATE_DST_PORT	228	2	Mapped destination port
Status and Event Fields			
FW_EVENT	233	1	High level event codes • 0—Ignore (invalid) • 1—Flow created • 2—Flow deleted • 3—Flow denied • 4—Flow alert
FW_EXT_EVENT	35,001	2	Extended event code. For normal records the length is 2 byte, and 4 byte for optional records.
Timestamp and Statistics Fields			
FW_EVENT_TIME_MSEC	323	8	Time, in milliseconds, (time since 0000 hours UTC [10] January 1, 1970) when the event occurred (if the event is a microevent, use 324 and 325, if it is a nanoevent)
FW_INITIATOR_OCTETS	231	4	Total number of Layer 4 payload bytes in the packet flow that arrives from the initiator
FW_RESPONDER_OCTETS	232	4	Total number of Layer 4 payload bytes in the packet flow that arrives from the responder

Field ID	Type	Length	Description
AAA Fields			
FW_USERNAME	40,000	20 or 64 depending on the template	AAA [11] user name
FW_USERNAME_MAX	40,000	64	AAA user name of the maximum permitted size
Alert Fields			
FW_HALFOPEN_CNT	35,012	4	Half-open session entry count
FW_BLACKOUT_SECS	35,004	4	Time, in seconds, when the destination is blacked out or unavailable
FW_HALFOPEN_HIGH	35,005	4	Configured maximum rate of TCP half-open session entries logged in one minute
FW_HALFOPEN_RATE	35,006	4	Current rate of TCP half-open session entries logged in one minute
FW_MAX_SESSIONS	35,008	4	Maximum number of sessions allowed for this zone pair or class ID
Miscellaneous			
FW_ZONEPAIR_ID	35,007	4	Zone pair ID
FW_CLASS_ID	51	4	Class ID
FW_ZONEPAIR_NAME	35,009	64	Zone pair name
FW_CLASS_NAME	100	64	Class name
FW_EXT_EVENT_DESC	35,010	32	Extended event description
FLOW_FIELD_CTS_SRC_GROUP_TAG	34000	2	Cisco Trustsec source tag
FW_SUMMARY_PKT_CNT	35,011	4	Number of packets represented by the drop/pass summary record
FW_EVENT_LEVEL	33003	4	Defines the level of the logged event • 0x01—Per box • 0x02—VRF • 0x03—Zone • 0x04—Class map • Other values are undefined

Field ID	Type	Length	Description
FW_EVENT_LEVEL_ID	33,004	4	Defines the identifier for the FW_EVENT_LEVEL field • If FW_EVENT_LEVEL is 0x02 (VRF), this field represents VRF_ID. • If FW_EVENT_LEVEL is 0x03 (zone), this field represents ZONE_ID. • If FW_EVENT_LEVEL is 0x04 (class map), this field represents CLASS_ID. • In all other cases the field ID will be 0 (zero). If FW_EVENT_LEVEL is not present, the value of this field must be zero.
FW_CONFIGURED_VALUE	33,005	4	Value that represents the configured half-open, aggressive-aging, and event-rate monitoring limit. The interpretation of this field value depends on the associated FW_EXT_EVENT field.
FW_ERM_EXT_EVENT	33,006	2	Extended event-rate monitoring code
FW_ERM_EXT_EVENT_DESC	33,007	N (string)	Extended event-rate monitoring event description string

[7] Internet Control Message Protocol
[8] Simple Network Management Protocol
[9] virtual routing and forwarding
[10] Coordinated Universal Time
[11] Authentication, Authorization, and Accounting

HSL Messages

The following are sample syslog messages from an Cisco ASR 1000 Series Aggregation Services Router:

Table 194: Syslog Messages and Their Templates

Message Identifier	Message Description	HSL Template
FW-6-DROP_PKT Type: Info	Dropping %s pkt from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s %s with ip ident %u %s %s Explanation: Packet dropped by firewall inspection. %s: tcp/udp/icmp/unknown prot/L7 prot %s:interface %CA:%u ip/ip6 addr: port %s:%s: zone pair name/ class name %s "due to" %s: fw_ext_event name %u ip ident %s: if tcp, tcp seq/ack number and tcp flags %s: username	FW_TEMPLATE_DROP_V4 or FW_TEMPLATE_DROP_V6

Message Identifier	Message Description	HSL Template
FW-6-SESS_AUDIT_TRAIL_START Type: Info	(target:class)-(%s:%s):Start %s session: initiator (%CA:%u) -- responder (%CA:%u) from %s %s %s Explanation: Start of an inspection session. This message is issued at the start of each inspection session and it records the source/destination addresses and ports. %s:%s: zonepair name: class name %s: l4/l7 protocolname %CA:%u ip/ip6 addr: port %s : interface %s : username %s : TODO Actual log: *Jan 21 20:13:01.078: %IOSXE-6-PLATFORM: F0: cpp_cp: CPP:00 Thread:125 TS:00000010570290947309 %FW-6-SESS_AUDIT_TRAIL_START: Start tcp session: initiator (10.1.1.1:43365) -- responder (10.3.21.1:23) from FastEthernet0/1/0	FW_TEMPLATE_START_AUDIT_V4 or FW_TEMPLATE_START_AUDIT_V6

Message Identifier	Message Description	HSL Template
FW-6-SESS_AUDIT_TRAIL Type: Info	(target:class)-(%s:%s):Stop %s session: initiator (%CA:%u) sent %u bytes -- responder (%CA:%u) sent %u bytes , from %s %s Explanation: Per-session transaction log of network activities. This message is issued at the end of each inspection session, and it records the source/destination addresses and ports, and the number of bytes transmitted by the client and the server. %s:%s: zonepair name: class name %s: l4/l7 protocolname %CA:%u ip/ip6 addr: port %u bytes counters %s: interface %s : TODO Actual log: *Jan 21 20:13:15.889: %IOSXE-6-PLATFORM: F0: cpp_cp: CPP:00 Thread:036 TS:00000010585102587819 %FW-6-SESS_AUDIT_TRAIL: Stop tcp session: initiator (10.1.1.1:43365) sent 35 bytes -- responder (11.1.1.1:23) sent 95 bytes, from FastEthernet0/1/0	FW_TEMPLATE_STOP_AUDIT_V4 or FW_TEMPLATE_STOP_AUDIT_V6
FW-4-UNBLOCK_HOST Type: Warning	(target:class)-(%s:%s):New TCP connections to host %CA no longer blocked Explanation: New TCP connection attempts to the specified host are no longer blocked. This message indicates that the blocking of new TCP connection attempts to the specified host has been removed. %s:%s: zonepair name: class name %CA: ip/ip6 addr	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_UNBLOCK_HOST

Message Identifier	Message Description	HSL Template
FW4HOST_TCP_ALERT_ON Type: Warning	"(target:class)-(%s:%s):Max tcp half-open connections (%u) exceeded for host %CA. Explanation: Exceeded the max-incomplete host limit for half-open TCP connections. This message indicates that a high number of half-open connections is coming to a protected server, and this may indicate that a SYN flood attack is in progress. %s:%s: zonepair name: class name %u: half open cnt %CA: ip/ip6 addr	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_HOST_TCP_ALERT_ON
FW-2-BLOCK_HOST Type: Critical	(target:class)-(%s:%s):Blocking new TCP connections to host %CA for %u minute%s (half-open count %u exceeded). Explanation: Exceeded the max-incomplete host threshold for TCP connections. Any subsequent new TCP connection attempts to the specified host is denied, and the blocking option is configured to block all subsequent new connections. The blocking will be removed when the configured block time expires. %s:%s: zonepair name: class name %CA: ip/ip6 addr %u blackout min %s: s if > 1 min blackout time %u: half open counter	FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V4 or FW_TEMPLATE_ALERT_TCP_HALF_OPEN_V6 with fw_ext_event id: FW_EXT_ALERT_BLOCK_HOST

Message Identifier	Message Description	HSL Template
FW-4-ALERT_ON Type: Warning	(target:class)-(%s:%s):%s, count (%u/%u) current rate: %u Explanation : Either the max-incomplete high threshold of half-open connections or the new connection initiation rate has been exceeded. This error message indicates that an unusually high rate of new connections is coming through the firewall, and a DOS attack may be in progress. This message is issued only when the max-incomplete high threshold is crossed. %s:%s: zonepair name: class name %s: "getting aggressive" %u/%u halfopen cnt/high %u: current rate	FW_TEMPLATE_ALERT_HALFOPEN_V4 or FW_TEMPLATE_ALERT_HALFOPEN_V6: with fw_ext_event id FW_EXT_SESS_RATE_ALERT_ON
FW-4-ALERT_OFF Type: Warning	(target:class)-(%s:%s):%s, count (%u/%u) current rate: %u Explanation: Either the number of half-open connections or the new connection initiation rate has gone below the max-incomplete low threshold. This message indicates that the rate of incoming new connections has slowed down and new connections are issued only when the max-incomplete low threshold is crossed. %s:%s: zonepair name: class name %s: "calming down" %u/%u halfopen cnt/high %u: current rate	FW_TEMPLATE_ALERT_HALFOPEN_V4 or FW_TEMPLATE_ALERT_HALFOPEN_V6: with fw_ext_event id FW_EXT_SESS_RATE_ALERT_OFF

Message Identifier	Message Description	HSL Template
FW-4-SESSIONS_MAXIMUM Type: Warning	Number of sessions for the firewall policy on "(target:class)-(%s:%s) exceeds the configured sessions maximum value %u Explanation: The number of established sessions have crossed the configured sessions maximum limit. %s:%s: zonepair name: class name %u: max session	FW_TEMPLATE_ALERT_MAX_SESSION
FW-6-PASS_PKT Type: Info	Passing %s pkt from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s %s with ip ident %u Explanation: Packet is passed by firewall inspection. %s: tcp/udp/icmp/unknown prot %s:interface %CA:%u src ip/ip6 addr: port %CA:%u dst ip/ip6 addr: port %s:%s: zonepair name: class name %s %s: "due to", "PASS action found in policy-map" %u: ip ident	FW_TEMPLATE_PASS_V4 or FW_TEMPLATE_PASS_V6
FW-6-LOG_SUMMARY Type: Info	%u packet%s %s from %s %CA:%u => %CA:%u (target:class)-(%s:%s) %s Explanation : Log summary for the number of packets dropped/passed %u %s: pkt_cnt, "s were" or "was" %s: "dropped"/ "passed" %s: interface %CA:%u src ip/ip6 addr: port %CA:%u dst ip/ip6 addr: port %s:%s: zonepair name: class name %s: username	FW_TEMPLATE_SUMMARY_V4 or FW_TEMPLATE_SUMMARY_V6 with FW_EVENT: 3 - drop 4 - pass

Firewall Extended Events

The event name of the firewall extended event maps the firewall extended event value to an event ID. Use the event name option record to obtain the mapping between an event value and an event ID.

Extended events are not part of standard firewall events (inspect, pass, or drop).

The following table describes the firewall extended events applicable prior to Cisco IOS XE Release 3.9S.

Table 195: Firewall Extended Events and Event Descriptions for Releases earlier than Cisco IOS XE Release 3.9S

Value	Event ID	Description
0	FW_EXT_LOG_NONE	No specific extended event.
1	FW_EXT_ALERT_UNBLOCK_HOST	New TCP connection attempts to the specified host are no longer blocked.
2	FW_EXT_ALERT_HOST_TCP_ALERT_ON	Maximum incomplete host limit for half-open TCP connections are exceeded.
3	FW_EXT_ALERT_BLOCK_HOST	All subsequent new TCP connection attempts to the specified host are denied because the maximum incomplete host threshold of half-open TCP connections is exceeded, and the blocking option is configured to block subsequent new connections.
4	FW_EXT_SESS_RATE_ALERT_ON	Maximum incomplete high threshold of half-open connections is exceeded, or the new connection initiation rate is exceeded.
5	FW_EXT_SESS_RATE_ALERT_OFF	Number of half-open TCP connections is below the maximum incomplete low threshold, or the new connection initiation rate has gone below the maximum incomplete low threshold.
6	FW_EXT_RESET	Reset connection.
7	FW_EXT_DROP	Drop connection.
10	FW_EXT_L4_NO_NEW_SESSION	No new session is allowed.
12	FW_EXT_L4_INVALID_SEG	Invalid TCP segment.
13	FW_EXT_L4_INVALID_SEQ	Invalid TCP sequence number.
14	FW_EXT_L4_INVALID_ACK	Invalid TCP acknowledgment (ACK).
15	FW_EXT_L4_INVALID_FLAGS	Invalid TCP flags.
16	FW_EXT_L4_INVALID_CHKSM	Invalid TCP checksum.
18	FW_EXT_L4_INVALID_WINDOW_SCALE	Invalid TCP window scale.

Value	Event ID	Description
19	FW_EXT_L4_INVALID_TCP_OPTIONS	Invalid TCP options.
20	FW_EXT_L4_INVALID_HDR	Invalid Layer 4 header.
21	FW_EXT_L4_OOO_INVALID_SEG	OoO [12] invalid segment.
24	FW_EXT_L4_SYNFLOOD_DROP	Synchronized (SYN) flood packets are dropped.
25	FW_EXT_L4_SCB_CLOSED	Session is closed while receiving packets.
26	FW_EXT_L4_INTERNAL_ERR	Firewall internal error.
27	FW_EXT_L4_OOO_SEG	OoO segment.
28	FW_EXT_L4_RETRANS_INVALID_FLAGS	Invalid retransmitted packet.
29	FW_EXT_L4_SYN_IN_WIN	Invalid SYN flag.
30	FW_EXT_L4_RST_IN_WIN	Invalid reset (RST) flag.
31	FW_EXT_L4_STRAY_SEG	Stray TCP segment.
32	FW_EXT_L4_RST_TO_RESP	Sending reset message to the responder.
33	FW_EXT_L4_CLOSE_SCB	Closing a session.
34	FW_EXT_L4_ICMP_INVAL_RET	Invalid ICMP [13] packet.
37	FW_EXT_L4_MAX_HALFSESSION	Maximum half-open session limit is exceeded.
38	FW_EXT_NO_RESOURCE	Resources (memory) are not available.
40	FW_EXT_INVALID_ZONE	Invalid zone.
41	FW_EXT_NO_ZONE_PAIR	Zone pairs are not available.
42	FW_EXT_NO_TRAFFIC_ALLOWED	Traffic is not allowed.
43	FW_EXT_FRAGMENT	Packet fragments are dropped.
44	FW_EXT_PAM_DROP	PAM [14] action is dropped.
45	FW_EXT_NOT_INITIATOR	Not a session-initiating packet. Occurs due to one of the following reasons: • If the protocol is TCP, the first packet is not a SYN packet. • If the protocol is ICMP, the first packet is not an ECHO or a TIMESTAMP packet.

Value	Event ID	Description
48	FW_EXT_ICMP_ERROR_PKTS_BURST	ICMP error packets came in burst mode. In burst mode, packets are sent repeatedly without waiting for a response from the responder interface.
49	FW_EXT_ICMP_ERROR_MULTIPLE_UNREACH	More than one ICMP error of type "destination unreachable" is received.
50	FW_EXT_ICMP_ERROR_L4_INVALID_SEQ	Embedded packet in the ICMP error message has an invalid sequence number.
51	FW_EXT_ICMP_ERROR_L4_INVALID_ACK	Embedded packet in the ICMP error message has an invalid acknowledge (ACK) number.
52	FW_EXT_MAX	Never used.

[12] Out-of-Order
[13] Internet Control Message Protocol
[14] Port-to-Application Mapping

The following table describes the firewall extended events from that are applicable to Cisco IOS XE Release 3.9S and later releases.

Table 196: Firewall Extended Events and Event Descriptions for Cisco IOS XE Release 3.9S and Later Releases

Value	Event ID	Description
0	FW_EXT_LOG_NONE	No specific extended event.
1	FW_EXT_FW_DROP_L4_TYPE_INVALID_HDR	Small datagram that cannot contain the Layer 4 ICMP, TCP, or UDP headers.
2	FW_EXT_FW_DROP_L4_TYPE_INVALID_ACK_FLAG	Did not contain an ACK flag, or a RST flag was set in the SYN/ACK packet during the TCP three-way handshake and the packet had an invalid sequence number.
3	FW_EXT_FW_DROP_L4_TYPE_INVALID_ACK_NUM	Occurs due to one of the following reasons: • When a packet's ACK value is less than the connection's oldest unacknowledged sequence number. • When a packet's ACK value is greater than the connection's next sequence number. • For SYN/ACK or ACK packets received during the three-way handshake, the sequence number is not equal to the initial sequence number plus 1.

Value	Event ID	Description
4	FW_EXT_FW_DROP_L4_TYPE_INVALID _TCP_INITIATOR	The first packet of a flow was not a SYN packet.
5	FW_EXT_FW_DROP_L4_TYPE_SYN _WITH_DATA	The SYN packet contains the payload and these SYN packet is not supported.
6	FW_EXT_FW_DROP_L4_TYPE_INVALI D_TCP_WIN_SCALE_OPTION	Invalid length for the TCP window-scale option.
7	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_SYNSENT_STATE	An invalid TCP segment was received in the SYNSENT state. Occurs due to one of the following reasons: • SYN/ACK has a payload. • SYN/ACK has other flags (push [PSH], urgent [URG], finish [FIN]) set. • Retransmit SYN message with a payload or invalid TCP flags (ACK, PSH, URG, FIN, RST) was received. • A non-SYN packet was received from the initiator.
8	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_SYNRCVD_STATE	A retransmitted SYN packet contains a payload or received a packet from the responder.
9	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PKT_TOO_OLD	Packet is older (lesser than) than the receiver's current TCP window.
10	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PKT_WIN_OVERFLOW	The sequence number of the packet is outside (greater than) the receiver's TCP window.
11	FW_EXT_FW_DROP_L4_TYPE_INVALID _SEG_PYLD_AFTER_FIN_SEND	A packet containing a payload was received from the sender after a FIN message was received.
12	FW_EXT_FW_DROP_L4_TYPE_INVALID_FLAGS	TCP flags associated with the packet are not valid. This may occur for the following reasons: • Extra flags along with the SYN flag, are set in the initial packet. Only the SYN flag is allowed in the initial packet. • Expected SYN/ACK did not contain a SYN flag, or the SYN/ACK contained extraneous flags in the second packet of the three-way handshake.

Value	Event ID	Description
13	FW_EXT_FW_DROP_L4_TYPE_INVALID_SEQ	Invalid sequence number. Occurs due to one of the following reasons: • The sequence number is less than the ISN [15]. • The sequence number is equal to the ISN but not equal to a SYN packet. • If the receive window size is zero and the packet contains data, or if the sequence number is greater than the last ACK number. • Sequence number falls beyond the TCP window.
14	FW_EXT_FW_DROP_L4_TYPE_RETRANS _INVALID_FLAGS	A retransmitted packet was already acknowledged by the receiver.
15	FW_EXT_FW_DROP_L4_TYPE_L7_OOO _SEG	The packet contains a TCP segment that arrived prior to the expected next segment.
16	FW_EXT_FW_DROP_L4_TYPE_SYN _FLOOD_DROP	Maximum-incomplete sessions configured for the policy have been exceeded and the host is in block time.
17	FW_EXT_FW_DROP_L4_TYPE_MAX _HALFSESSION	Exceeded the number of allowed half-open sessions.
18	FW_EXT_FW_DROP_L4_TYPE_ TOO_MANY_PKTS	Exceeded the maximum number of simultaneous inspectable packets allowed per flow. The number is currently set to allow 25 simultaneous packets to be inspected. The simultaneous inspection prevents any one flow from monopolizing more than its share of processor resources.
19	FW_EXT_FW_DROP_L4_TYPE_TOO _MANY_ICMP_ERR_PKTS	Exceeded the maximum number of ICMP error packets allowed per flow. This log is triggered by the firewall base inspection.
20	FW_EXT_FW_DROP_L4_TYPE_UNEXPECT _TCP_PYLD	Retransmitted SYN/ACK from the responder included a payload. Payloads are not allowed during a TCP three-way handshake negotiation.
21	FW_EXT_FW_DROP_L4_TYPE_INTERNAL _ERR_UNDEFINED_DIR	Packet direction is undefined.

Value	Event ID	Description
22	FW_EXT_FW_DROP_L4_TYPE_SYN _IN_WIN	A TCP packet of an established session arrived with the SYN flag set. A SYN flag is not allowed after the initial two packets of the three-way handshake.
23	FW_EXT_FW_DROP_L4_TYPE_RST _IN_WIN	A TCP packet with the RST flag set was received with a sequence number that is outside the last received acknowledgment. The packet may be sent out of order.
24	FW_EXT_FW_DROP_L4_TYPE_ STRAY_SEG	An unexpected packet was received after the flow was torn down, or a packet was received from the responder before the initiator sent a valid SYN flag.
25	FW_EXT_FW_DROP_L4_TYPE_ RST_TO_RESP	A SYN/ACK flag was expected from the responder. However, a packet with an invalid sequence number was received. The zone-based firewall sent a RST flag to the responder.
26	FW_EXT_FW_DROP_L4_TYPE _I NTERNAL_ERR_ICMP_NO_NAT	The ICMP packet is NAT [16] translated; but internal NAT information is missing. An internal error.
27	FW_EXT_FW_DROP_L4_TYPE _ INTERNAL_ERR_ICMP_ALLOC_FAIL	Failed to allocate an ICMP error packet during an ICMP inspection.
28	FW_EXT_FW_DROP_L4_TYPE _INTERNAL_ERR_ICMP_GET_STAT_BLK_FAIL	The classification result did not have the required statistics memory. The policy information was not properly downloaded to the data plane.
29	FW_EXT_FW_DROP_L4_TYPE _INTERNAL_ERR_ICMP_DIR_NOT_IDENTIFIED	Packet direction is not defined.
30	FW_EXT_FW_DROP_L4_TYPE_ ICMP_SCB_CLOSE	Received an ICMP packet while the session is being torn down.
31	FW_EXT_FW_DROP_L4_TYPE_ ICMP_PKT_NO_IP_HDR	No IP header in the payload of the ICMP error packet.
32	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR_NO _IP_NO_ICMP	The ICMP error packet has no IP or ICMP, which is probably due to a malformed packet.
33	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _PKTS_BURST	The ICMP error packet exceeded the burst limit of 10
34	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _MULTIPLE_UNREACH	The ICMP error packet exceeded the “Unreachable” limit. Only the first unreachable packet is allowed to pass.

Value	Event ID	Description
35	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _L4_INVALID_SEQ	The sequence number of the embedded packet does not match the sequence number of the TCP packet that triggers the ICMP error packet.
36	FW_EXT_FW_DROP_L4_TYPE_ ICMP_ERROR _L4_INVALID_ACK	The TCP packet contained in an ICMP error packet payload has an ACK flag that was not seen before.
37	FW_EXT_FW_DROP_L4_TYPE_ICMP_PKT_TOO _SHORT	The ICMP error packet length is less than the IP header length plus the ICMP header length.
38	FW_EXT_FW_DROP_L4_TYPE_ SESSION_LIMIT	Resources exceeded the session limit while promoting for an imprecise channel.
39	FW_EXT_FW_DROP_L4_TYPE_ SCB_CLOSE	A TCP packet was received on a closed session.
40	FW_EXT_FW_DROP_INSP_TYPE_ POLICY_NOT_PRESENT	A policy is not present in a zone pair.
41	FW_EXT_FW_DROP_INSP_TYPE_ SESS_MISS_POLICY _NOT_PRESENT	A zone pair is configured in the same zone, but the zone does not have any policies.
44	FW_EXT_FW_DROP_INSP_TYPE_ CLASS_ACTION_DROP	The classification action is to drop the non-ICMP, TCP, and UDP packets.
45	FW_EXT_FW_DROP_INSP_TYPE_ PAM_LOOKUP_FAIL	The classification action is to drop the PAM entry.
48	FW_EXT_FW_DROP_INSP_TYPE_ INTERNAL_ERR_ GET_STAT_BLK_FAIL	Failed to get the statistic block from the classification result bytes.
49	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_SYNCOOKIE _MAX_DST	The maximum entry limit for SYN flood packets is reached.
50	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_INTERNAL _ERR_ALLOC_FAIL	Cannot allocate memory for the destination table entry.
51	FW_EXT_FW_DROP_SYNCOOKIE_ TYPE_SYN_COOKIE _TRIGGER	The SYN cookie logic is triggered. Indicates that the SYN/ACK with the SYN cookie was sent and the original SYN packet was dropped.
52	FW_EXT_FW_DROP_POLICY_ TYPE_FRAG_DROP	The first fragment of a VFR [17] packet is dropped and all associated remaining fragments will be dropped.
53	FW_EXT_FW_DROP_POLICY_ TYPE_ACTION_DROP	The classification action is to drop the packet.

Value	Event ID	Description
54	FW_EXT_FW_DROP_POLICY_ TYPE_ICMP_ACTION_DROP	The policy action of the ICMP embedded packet is DROP.
55	FW_EXT_FW_DROP_L7_TYPE_ NO_SEG	Layer 7 ALG [18] does not inspect inspect-segmented packets.
56	FW_EXT_FW_DROP_L7_TYPE_ NO_FRAG	Layer 7 ALG does not inspect fragmented packets.
57	FW_EXT_FW_DROP_L7_TYPE_ UNKNOWN_PROTO	Unknown application protocol type.
58	FW_EXT_FW_DROP_L7_TYPE_ ALG_RET_DROP	Layer 7 ALG inspection resulted in a packet drop.
59	FW_EXT_FW_DROP_NONSESSION _TYPE	Session creation has failed.
60	FW_EXT_FW_DROP_NO_NEW _SESSION_TYPE	During initial HA [19] states, a new session is not allowed.
61	FW_EXT_FW_DROP_NOT_ INITIATOR_TYPE	Not a session initiator packet.
62	FW_EXT_FW_DROP_INVALID _ZONE_TYPE	When default zones are not enabled, traffic is only allowed between interfaces that are associated with security zones.
64	FW_EXT_FW_DROP_NO_ FORWARDING_TYPE	The firewall is not configured.
65	FW_EXT_FW_DROP_ BACKPRESSURE_TYPE	The firewall backpressure can be enabled if HSL [20] is enabled, and the HSL logger was unable to send a log message. Backpressure will remain enabled until HSL is able to send a log.
66	FW_EXT_FW_DROP_L4_TYPE_INTERNAL _ERR _SYNFLOOD_ALLOC_HOSTDB_FAIL	During SYN processing, host rate limits are tracked. The host entry could not be allocated.
67	FW_EXT_FW_DROP_L4_TYPE_ SYNFLOOD_BLACKOUT_DROP	If the configured half-open connection limit is exceeded and blackout time is configured, all new connections to the specified IP address are dropped.
68	FW_EXT_FW_DROP_L7_TYPE_ PROMOTE_FAIL_NO_ZONE_PAIR	A failed policy. When an ALG attempts to promote a session because no zone pairs are configured, the policy fails.
69	FW_EXT_FW_DROP_L7_TYPE_ PROMOTE_FAIL_NO_POLICY	A failed policy. When an ALG attempts to promote a session due to no policy, the policy fails.

Value	Event ID	Description
	FW_EXT_FW_DROP_L4_TYPE_ONEFW _SCB_CLOSE	A packet is received after the Context-Aware firewall (CXSC) requested a teardown.
	FW_EXT_FW_DROP_L4_TYPE_ONEFW _FAIL_CLOSE	CXSC is not running.

[15] initial sequence number
[16] Network Address Translation
[17] virtual fragmentation and reassembly
[18] application layer gateway
[19] high availability
[20] high-speed logging

How to Configure Firewall High-Speed Logging

Enabling High-Speed Logging for Global Parameter Maps

By default, high-speed logging (HSL) is not enabled and firewall logs are sent to a logger buffer located in the Route Processor (RP) or the console. When HSL is enabled, logs are sent to an off-box, high-speed log collector. Parameter maps provide a means of performing actions on the traffic that reaches a firewall and a global parameter map applies to the entire firewall session table. Perform this task to enable high-speed logging for global parameter maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **log dropped-packets**
5. **log flow-export v9 udp destination** *ip-address port-number*
6. **log flow-export template timeout-rate** *seconds*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**parameter-map type inspect global** **Example:** `Device(config)# parameter-map type inspect global`	Configures a global parameter map and enters parameter-map type inspect configuration mode.
Step 4	**log dropped-packets** **Example:** `Device(config-profile)# log dropped-packets`	Enables dropped-packet logging.
Step 5	**log flow-export v9 udp destination** *ip-address port-number* **Example:** `Device(config-profile)# log flow-export v9 udp destination 10.0.2.0 5000`	Enables NetFlow event logging and provides the IP address and the port number of the log collector.
Step 6	**log flow-export template timeout-rate** *seconds* **Example:** `Device(config-profile) log flow-export template timeout-rate 5000`	Specifies the template timeout value.
Step 7	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Enabling High-Speed Logging for Firewall Actions

Perform this task enable high-speed logging if you have configured inspect-type parameter maps. Parameter maps specify inspection behavior for the firewall and inspection parameter-maps for the firewall are configured as the inspect type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **audit-trail on**
5. **alert on**
6. **one-minute** {**low** *number-of-connections* | **high** *number-of-connections*}
7. **tcp max-incomplete host** *threshold*
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect** *parameter-map-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect parameter-map-hsl`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword, and enters parameter-map type inspect configuration mode.
Step 4	**audit-trail on** **Example:** `Device(config-profile)# audit-trail on`	Enables audit trail messages. • You can enable audit-trail to a parameter map to record the start, stop, and duration of a connection or session, and the source and destination IP addresses.
Step 5	**alert on** **Example:** `Device(config-profile)# alert on`	Enables stateful-packet inspection alert messages that are displayed on the console.
Step 6	**one-minute** {**low** *number-of-connections* | **high** *number-of-connections*} **Example:** `Device(config-profile)# one-minute high 10000`	Defines the number of new unestablished sessions that cause the system to start deleting half-open sessions and stop deleting half-open sessions.
Step 7	**tcp max-incomplete host** *threshold* **Example:** `Device(config-profile)# tcp max-incomplete host 100`	Specifies the threshold and blocking time values for TCP host-specific, denial of service (DoS) detection and prevention.
Step 8	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect policy-map-hsl`	Creates an inspect-type policy map and enters policy map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:**	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.

	Command or Action	Purpose
	`Device(config-pmap)# class type inspect class-map-tcp`	
Step 11	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect parameter-map-hsl`	(Optional) Enables stateful packet inspection.
Step 12	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuration Examples for Firewall High-Speed Logging

Example: Enabling High-Speed Logging for Global Parameter Maps

The following example shows how to enable logging of dropped packets, and to log error messages in NetFlow Version 9 format to an external IP address:

```
Device# configure terminal
Device(config)# parameter-map type inspect global
Device(config-profile)# log dropped-packets
Device(config-profile)# log flow-export v9 udp destination 10.0.2.0 5000
Device(config-profile)# log flow-export template timeout-rate 5000
Device(config-profile)# end
```

Example: Enabling High-Speed Logging for Firewall Actions

The following example shows how to configure high-speed logging (HSL) for inspect-type parameter-map parameter-map-hsl.

```
Device# configure terminal
Device(config)# parameter-map type inspect parameter-map-hsl
Device(config-profile)# audit trail on
Device(config-profile)# alert on
Device(config-profile)# one-minute high 10000
Device(config-profile)# tcp max-incomplete host 100
Device(config-profile)# exit
Device(config)# poliy-map type inspect policy-map-hsl
Device(config-pmap)# class type inspect class-map-tcp
Device(config-pmap-c)# inspect parameter-map-hsl
Device(config-pmap-c)# end
```

Additional References for Firewall High-Speed Logging

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 144

TCP Reset Segment Control

The TCP Reset Segment Control feature provides a mechanism to configure if a TCP reset (RST) segment should be sent when a session deletion occurs for half-close, half-open, or idle sessions.

Information about TCP Reset Segment Control

TCP Reset Segment Control

The TCP header contains a flag known as the reset (RST) flag. A TCP segment is sent with the RST flag whenever a segment arrives that does not meet the criteria for a referenced connection. For example, a TCP segment is sent with a RST flag when a connection request is received on the destination port, but no process is listening at that port.

This behavior is defined in RFC 793, Transmission Control Protocol, for host-to-host communication and implemented by various vendors. However, for the network devices that reside on the network between hosts, specific rules have not been defined to determine if the device should send the TCP RST segment to the connection initiator, receiver, or both when sessions (half-open, idle, half-close) are cleared. Some devices send the TCP RST segment to both sender and receiver ports when a session is cleared, while some devices silently remove the session in the session table without sending out any TCP RST segments.

The TCP Reset Segment Control feature provides a mechanism to configure if a TCP RST segment should be sent when a session is cleared for half-close, half-open, or idle sessions.

A half-open session is an unestablished session initiated by a TCP synchronization (SYN) segment but is incomplete as only a TCP three-way handshake occurs and a timer is started.

TCP provides the ability for one end of a connection to terminate its output while still receiving data from the other end of the connection. This TCP state is called the half-close state. A session enters the half-close state when it receives the first TCP FIN segment and starts a timer. If another segment is received before the session timeout occurs, then the timer is restarted.

You can set the timeout value for half-open and half-close sessions by using the **tcp synwait-time** and **tcp finwait-time** commands respectively. The default timeout value is 30 seconds.

An idle session is a TCP session that is active between two devices and no data is transmitted by either of the devices for a prolonged period of time. You can set the timeout value for an idle session by using the **tcp idle-time** command. The default timeout value for idle sessions is 3600 seconds.

Once the timeout occurs on the TCP sessions and the session is cleared, the TCP RST segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

How to Configure TCP Reset Segment Control

Configuring TCP Reset for Half-Open Sessions

A half-open session is an unestablished session that is initiated by a TCP synchronization (SYN) segment but has an incomplete three-way handshake. A timer is started as soon as the incomplete three-way handshake occurs. You can set the timer values for a half-open session timeout by using the **tcp synwait-time** command. The default timeout value for these sessions is 30 seconds.

When the timeout occurs and the session is cleared on the half-open TCP session, the TCP reset (RST) segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

If you configure the **tcp half-open reset on** command, the TCP RST segment is sent to both ends of the half-open session when the session is cleared. If you configure the **tcp half-open reset off** command, the TCP RST segment is not transmitted when the session is cleared.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp synwait-time** *seconds*
5. **tcp half-open reset** {**off** | **on**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	(Optional) Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.

	Command or Action	Purpose
Step 4	**tcp synwait-time** *seconds* **Example:** `Device(config-profile)# tcp synwait-time 10`	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 5	**tcp half-open reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp half-open reset on`	Specifies whether the TCP RST segment should be sent when timeout occurs and the session is cleared for a half-open session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuring TCP Reset for Half-Close Sessions

TCP provides the ability for one end of a connection to terminate its output, while still receiving data from the other end of the connection. This TCP state is called the half-close state. A session enters the half-close state when it receives the first TCP finish (FIN) segment and starts a timer. If another segment is received before the session timeout occurs, then the timer is restarted. You can set the timeout value for a half-close session by using the **tcp finwait-time** command. The default timeout value for half-close sessions is 30 seconds.

Once the timeout occurs on the half-close TCP session, the TCP RST segment is sent and the session will be reset only if the TCP reset segment control is configured on the sessions.

If you configure the **tcp half-close reset on** command, the TCP RST segment is sent to both ends of the half-open session when timeout occurs and the session is cleared. If you configure the **tcp half-close reset off** command, the TCP RST segment is not transmitted when the session timeout occurs and the session is cleared.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp finwait-time** *seconds*
5. **tcp half-close reset** {**off** | **on**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.
Step 4	**tcp finwait-time** *seconds* **Example:** `Device(config-profile)# tcp finwait-time 10`	(Optional) Specifies how long a TCP session will be managed after the firewall detects a FIN-exchange.
Step 5	**tcp half-close reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp half-close reset on`	Specifies whether the TCP RST segment should be sent when session deletion occurs on a half-open session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuring TCP Reset for Idle Sessions

An idle session is a TCP session that is active between two devices and no data is transmitted by either device for a prolonged period of time. You can set the timeout value for an idle session by using the **tcp idle-time** command. The default timeout value for idle sessions is 3600 seconds.

Once the timeout occurs on the idle TCP session, the TCP RST segment is sent and the session will be reset if the TCP reset segment control is configured on the session.

If you configure the **tcp idle reset on** command, the TCP RST segment is sent to both ends of the idle session when timeout occurs and the session is cleared. If you configure the **tcp idle reset off** command, the TCP RST segment is not transmitted when the session timeout occurs and the session is cleared.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **tcp idle-time** *seconds*
5. **tcp idle reset** {**off** | **on**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** *parameter-map-name* **Example:** `Device(config)# parameter-map type inspect pmap-name`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** keyword and enters parameter-map type inspect configuration mode.
Step 4	**tcp idle-time** *seconds* **Example:** `Device(config-profile)# tcp idle-time 90`	(Optional) Configures the timeout for TCP sessions.
Step 5	**tcp idle reset** {**off** | **on**} **Example:** `Device(config-profile)# tcp idle reset on`	Specifies whether the TCP RST segment should be sent when session deletion occurs on an idle session.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and enters privileged EXEC mode.

Configuration Examples for TCP Reset Segment Control

Example: Configuring TCP Reset for Half-Open Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
Device(config-profile)# tcp synwait-time 10
Device(config-profile)# tcp half-open reset on
Device(config-profile)# end
```

Example: Configuring TCP Reset for Half-Close Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
```

```
Device(config-profile)# tcp finwait-time 10
Device(config-profile)# tcp half-close reset on
Device(config-profile)# end
```

Example: Configuring TCP Reset for Idle Sessions

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-name
Device(config-profile)# tcp idle-time 90
Device(config-profile)# tcp idle reset on
Device(config-profile)# end
```

Additional References for TCP Reset Segment Control

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 793	Transmission Control Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for TCP Reset Segment Control

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 197: Feature Information for TCP Reset Segment Control

Feature Name	Releases	Feature Information
TCP Reset Segment Control	Cisco IOS XE Release 3.8S	The TCP Reset Segment Control feature provides a consistent mechanism to configure if the TCP RST bits should be sent out when a session is cleared for half-open, half-close, and idle sessions. The following commands were introduced or modified: **tcp idle reset**, **tcp half-close reset**, and **tcp half-open reset**.

CHAPTER 145

Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

The Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall feature disables the strict checking of the TCP window-scaling option in a firewall.

Information About Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Loose Checking Option for TCP Window Scaling Overview

TCP provides various TCP extensions to improve performance over high-bandwidth and high-speed data paths. One such extension is the TCP window-scaling option. The loose-checking option for TCP window-scaling turns off strict checking of the window-scaling option described in RFC 1323.

A larger window size is recommended to improve TCP performance in network paths with large bandwidth-delay product characteristics that are called Long Fat Networks (LFNs). TCP window scaling expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header. The window size can increase to a scale factor of 14. Typical applications use a scale factor of 3 when deployed in LFNs.

A firewall implementation enforces strict checking of the TCP window-scaling option. A firewall drops SYN/ACK packets that have the TCP window-scaling option if it was not offered in the initial synchronization (SYN) packet for the TCP three-way handshake. The window-scale option is sent only in a SYN segment, which is a segment with the SYN bit on. Therefore, the window scale is fixed in each direction when a connection is opened.

Use the **tcp window-scale-enforcement loose** command to disable the strict checking of the TCP window-scaling option in TCP SYN segments.

How to Configure Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Configuring the TCP Window-Scaling Option for a Firewall

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **tcp window-scale-enforcement loose**
5. **exit**
6. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
7. **match protocol** [*parameter-map*] [**signature**]
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** *class-map-name*
11. **inspect** [*parameter-map-name*]
12. **exit**
13. **class** *name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect** {*parameter-map-name*	**global**	**default**} **Example:** `Device(config)# parameter-map type inspect pmap-fw`	Configures an inspect parameter map and enters profile configuration mode.
Step 4	**tcp window-scale-enforcement loose** **Example:**	Disables the strict checking of the TCP window-scaling option in a firewall.

	Command or Action	Purpose
	`Device(config-profile)# tcp window-scale-enforcement loose`	
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits profile configuration mode and returns to global configuration mode.
Step 6	**class-map type inspect** {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any internet-traffic-class`	Creates an inspect-type class map and enters QoS class-map configuration mode.
Step 7	**match protocol** [*parameter-map*] [**signature**] **Example:** `Device(config-cmap)# match protocol tcp`	Configures a match criteria for a class map on the basis of the specified protocol.
Step 8	**exit** **Example:** `Device(config-cmap)# exit`	Exits the QoS class-map configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect private-internet-policy`	Creates an inspect-type policy map and enters QoS policy-map configuration mode.
Step 10	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect internet-traffic-class`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect pmap-fw`	Enables stateful packet inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and returns to QoS policy-map configuration mode.
Step 13	**class** *name* **Example:** `Device(config-pmap)# class class-default`	Associates the map class with a specified data-link connection identifier (DLCI).
Step 14	**end** **Example:** `Device(config-pmap)# end`	Exits QoS policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Zone and Zone Pair for a TCP Window Scaling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address*
5. **zone-member security** *security-zone-name*
6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address*
9. **zone-member security** *security-zone-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/5`	Specifies an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Assigns an interface IP address.
Step 5	**zone-member security** *security-zone-name* **Example:** `Device(config-if)# zone-member security private`	Configures the interface as a zone member.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/6`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ip address** *ip-address* **Example:** `Device(config-if)# ip address 209.165.200.225 255.255.255.0`	Assigns an IP address to an interface.
Step 9	**zone-member security** *security-zone-name* **Example:** `Device(config-if)# zone-member security internet`	Configures an interface as a zone member.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for TCP Window-Scaling

Example: Configuring the TCP Window-Scaling Option for a Firewall

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# tcp window-scale-enforcement loose
Device(config-profile)# exit
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
Device(config-pmap-c)#exit
Device(config-pmap)# class class-default
Device(config-pmap)#end
```

Example: Configuring a Zone and Zone Pair for TCP Window Scaling

```
Device# enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# zone-member security private
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/1/6
Device(config-if)# ip address 209.165.200.225 255.255.255.0
Device(config-if)# zone-member security internet
Device(config-if)# end
```

Feature Information for Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 198: Feature Information for Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall

Feature Name	Releases	Feature Information
Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall	Cisco IOS XE Release 3.10S	Loose Checking Option for TCP Window Scaling in Zone-Based Policy Firewall feature disables the strict checking of the TCP Window Scaling option in an IOS-XE firewall. The following command was introduced or modified: **tcp window-scale-enforcement loose**. In Cisco IOS XE Release 3.10S, support was added for the Cisco CSR 1000V Series Routers.

CHAPTER 146

Enabling ALGs and AICs in Zone-Based Policy Firewalls

Zone-based policy firewalls support Layer 7 application protocol inspection along with application-level gateways (ALGs) and application inspection and control (AIC). Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through a security module.

Prior to the introduction of Enabling ALGs and AICs in Zone-Based Policy Firewalls feature, the Layer 7 protocol inspection was automatically enabled along with the ALG/AIC configuration. With this feature you can enable or disable Layer 7 inspection by using the **no application-inspect** command.

This module provides an overview of the Enabling ALGs and AICs in Zone-Based Policy Firewalls feature and describes how to configure it.

Information About Enabling ALGs and AICs in Zone-Based Policy Firewalls

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Enabling Layer 7 Application Protocol Inspection Overview

Zone-based policy firewalls support Layer 7 protocol inspection along with application-level gateways (ALG) and application inspection and control (AIC). Layer 7 protocol inspection is automatically enabled along with the ALG/AIC configuration.

Layer 7 application protocol inspection is a technique that interprets or understands application-layer protocols and performs appropriate firewall or Network Address Translation (NAT) action. Certain applications require special handling of the data portion of a packet when the packet passes through the security module on a device. Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through the security module. Based on the configured traffic policy, the security module accepts or rejects packets to ensure the secure use of applications and services.

Sometimes, application inspection implementation issues can cause application packet drop and make networks unstable. Prior to the introduction of the Enabling ALGs and AICs in Zone-Based Policy Firewall feature, to disable application inspection you had to define an access control list (ACL) with the target Layer 7 protocol port define a class map that matches this ACL and matches either the TCP or UDP protocol to bypass the inspection for a specific Layer 7 protocol.

With the introduction of the Enabling ALGs and AICs in Zone-Based Policy Firewall feature, you can enable or disable Layer 7 protocol inspection for a specific protocol or for all supported Layer 7 protocols with the **application-inspect** command. Any configuration changes to a parameter map applies only to new sessions. For example, when you disable FTP Layer 7 inspection, the newly created sessions skip FTP Layer 7 inspection, while existing sessions before the configuration change will perform FTP Layer 7 inspection. For all sessions to perform the configuration change, you must delete all sessions and re-create them.

You can enable Layer 7 application protocol inspection for an individual parameter map or for a global firewall.

How to Enable ALGs and AICs in Zone-Based Policy Firewalls

Enabling Layer 7 Application Protocol Inspection on Firewalls

Application protocol inspection is enabled by default. Use the **no application-inspect** command to disable application protocol inspection.

Use the **application-inspect** command to reconfigure application protocol inspection, if you have disabled it for any reason. Configure either the **parameter-map type inspect** command or the **parameter-map type inspect-global** command before configuring the **application-inspect** command.

You can only configure either the **parameter-map type inspect** command or the **parameter-map type inspect-global** command at any time.

Use the

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. Do one of the following:
   - **parameter-map type inspect** *parameter-map-name*
   - **parameter-map type inspect-global**
4. **application-inspect** {**all** | *protocol-name*}
5. **exit**
6. **class-map type inspect** {**match-all** | **match-any**} *class-map-name*
7. **match protocol** *protocol-name*
8. **exit**
9. **policy-map type inspect** *policy-map-name*
10. **class type inspect** {*class-map-name* | **class-default**}
11. **inspect** *parameter-map-name*
12. **exit**
13. **class** {*class-map-name* | **class-default**}
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **parameter-map type inspect** *parameter-map-name*
	• **parameter-map type inspect-global**
	Example:
	`Device(config)# parameter-map type inspect pmap-fw`
	or
	`Device(config)# parameter-map type inspect-global`
		• (Optional) Enables an inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode.
		• (Optional) Enables a global parameter map and enters parameter-map type inspect configuration mode.

Step 4	**application-inspect** {**all**	*protocol-name*} **Example:** `Device(config-profile)# application-inspect msrpc`	Enables application inspection for the specified protocols.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	Exits parameter-map type inspect configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**class-map type inspect** {**match-all** | **match-any**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any internet-traffic-class`	Creates an inspect type class map and enters class map configuration mode.
Step 7	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol msrpc`	Configures a match criterion for a class map based on the specified protocol.
Step 8	**exit** **Example:** `Device(config-cmap)# exit`	Exits class map configuration mode and returns to global configuration mode.
Step 9	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect private-internet-policy`	Creates an inspect type policy map and enters policy map configuration mode.
Step 10	**class type inspect** {*class-map-name* | **class-default**} **Example:** `Device(config-pmap)# class type inspect internet-traffic-class`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 11	**inspect** *parameter-map-name* **Example:** `Device(config-pmap-c)# inspect pmap-fw`	Enables stateful packet inspection.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy map configuration mode.
Step 13	**class** {*class-map-name* | **class-default**} **Example:** `Device(config-pmap)# class class-default`	Specifies the default class so that you can configure or modify the policy.
Step 14	**end** **Example:** `Device(config-pmap)# end`	Exits policy map configuration mode and returns to privileged EXEC mode.

Configuring Zones for Enabling Layer 7 Application Protocol Inspection

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **zone security** {**default** | *security-zone*}
4. **exit**
5. **zone security** {**default** | *security-zone*}
6. **exit**
7. **zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *security-zone*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *security-zone*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {**default**	*security-zone*} **Example:** `Device(config)# zone security private`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • You need two security zones to create a zone pair: a source and a destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {**default**	*security-zone*} **Example:** `Device(config)# zone security internet`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**zone-pair security** *zone-pair* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security private-internet source private destination internet`	Creates a zone pair and enters security zone pair configuration mode.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy`	Attaches a firewall policy map to the destination zone pair. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *security-zone* **Example:** `Device(config-if)# zone-member security private`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2/2`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *security-zone* **Example:** `Device(config-if)# zone-member security internet`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Enabling ALGs and AICs in Zone-Based Policy Firewalls

Example: Enabling Layer 7 Application Protocol Inspection on Firewalls

The following example shows how to enable Layer 7 application protocol inspection after configuring the **parameter-map type inspect** command. You can enable application inspection after configuring the **parameter-map type inspect-global** command also.

You can only configure either the **parameter-map type inspect** or the **parameter-map type inspect-global** command at any time.

```
Device# configure terminal
Device(config)# parameter-map type inspect pmap-fw
Device(config-profile)# application-inspect msrpc
Device(config-profile)# exit
Device(config)# class-map type inspect match-any internet-traffic-class
Device(config-cmap)# match protocol msrpc
Device(config-cmap)# exit
Device(config)# policy-map type inspect private-internet-policy
Device(config-pmap)# class type inspect internet-traffic-class
Device(config-pmap-c)# inspect pmap-fw
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# end
```

Example: Configuring Zones for Enabling Layer 7 Application Protocol Inspection

```
Device# configure terminal
Device(config)# zone security private
Device(config-sec-zone)# exit
Device(config)# zone security internet
Device(config-sec-zone)# exit
Device(config)# zone-pair security private-internet source private destination internet
Device(config-sec-zone-pair)# service-policy type inspect private-internet-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# zone-member security private
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/2/2
Device(config-if)# zone-member security internet
Device(config-if)# end
```

Additional References for Enabling ALGs and AICs in Zone-Based Policy Firewalls

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Enabling ALGs and AICs in Zone-Based Policy Firewalls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 199: Feature Information for Enabling ALGs and AICs in Zone-Based Policy Firewalls

Feature Name	Releases	Feature Information
Enabling ALGs and AICs in Zone-Based Policy Firewalls	Cisco IOS XE Release 3.11S	Zone-based policy firewalls support Layer 7 application protocol inspection along with application-level gateways (ALGs) and application inspection and control (AIC). Layer 7 application protocol inspection helps to verify the protocol behavior and identify unwanted or malicious traffic that passes through security module. Prior to the introduction of Enabling ALGs and AICs in Zone-Based Policy Firewalls feature, the Layer 7protocol inspection was automatically enabled along with the ALG/AIC configuration. With this feature you can enable or disable Layer 7 inspection by using the no application-inspect command. In Cisco IOS XE Release 3.11S, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers, Cisco 4400 Series Integrated Services Routers, and Cisco Cloud Services Routers 1000V. The following commands were introduced or modified: **application-inspect**, **show parameter-map type inspect**, and **show platform software firewall**.

CHAPTER 147

Configuring Firewall TCP SYN Cookie

The Firewall TCP SYN Cookie feature protects your firewall from TCP SYN-flooding attacks. TCP SYN-flooding attacks are a type of denial-of-service (DoS) attack. Usually, TCP synchronization (SYN) packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or a program pretends to be another by falsifying data and thereby gaining an illegitimate advantage. TCP SYN-flooding can take up all resources on a firewall or an end host, thereby causing DoS to legitimate traffic. To prevent TCP SYN-flooding on a firewall and the end hosts behind the firewall, you must configure the Firewall TCP SYN Cookie feature.

Restrictions for Configuring Firewall TCP SYN Cookie

- Because a default zone does not support zone type parameter map, you cannot configure the Firewall TCP SYN Cookie feature for a default zone.
- The Firewall TCP SYN Cookie feature does not support per-subscriber firewall.

Information About Configuring Firewall TCP SYN Cookie

TCP SYN Flood Attacks

The Firewall TCP SYN Cookie feature implements software to protect the firewall from TCP SYN-flooding attacks, which are a type of DoS attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Because these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a website, accessing e-mail, using FTP service, and so on.

SYN flood attacks are divided into two types:

- Host flood—SYN flood packets are sent to a single host aiming to utilize all resources on that host.
- Firewall session table flood—SYN flood packets are sent to a range of addresses behind the firewall, with the aim of exhausting the session table resources on the firewall and thereby denying resources to the legitimate traffic going through the firewall.

The Firewall TCP SYN Cookie feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. The firewall intercepts TCP SYN packets that are sent from clients to servers. When the TCP SYN cookie is triggered, it acts on all SYN packets that are destined to the configured VPN Routing and Forwarding (VRF) or zone. The TCP SYN cookie establishes a connection with the client on behalf of the destination server and another connection with the server on behalf of the client and knits together the two half-connections transparently. Thus, connection attempts from unreachable hosts will never reach the server. The TCP SYN cookie intercepts and forwards packets throughout the duration of the connection.

The Firewall TCP SYN Cookie feature provides session table SYN flood protection for the global routing domain and for the VRF domain. Because the firewall saves sessions in a global table, you can configure a limit to the number of TCP half-opened sessions. A TCP half-opened session is a session that has not reached the established state. In a VRF-aware firewall, you can configure a limit to the number of TCP half-opened sessions for each VRF. At both the global level and at the VRF level, when the configured limit is reached, the TCP SYN cookie verifies the source of the half-opened sessions before creating more sessions.

How to Configure Firewall TCP SYN Cookie

Configuring Firewall Host Protection

TCP SYN packets are sent to a single host with the aim of taking over all resources on the host. You can configure host protection only for the source zone. Configuring protection on the destination zone will not protect the destination zone from TCP SYN attacks.

Perform this task to configure the firewall host protection.

Note You can specify the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-zone** *zone-pmap-name*
4. **tcp syn-flood rate per-destination** *maximum-rate*
5. **max-destination** *limit*
6. **exit**
7. **zone security** *zone-name*
8. **protection** *parameter-map-name*
9. **exit**
10. **show parameter-map type inspect-zone** *zone-pmap-name*

11. **show zone security**
12. **show policy-firewall stats zone** *zone-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-zone** *zone-pmap-name* **Example:** `Router(config)# parameter-map type inspect-zone zone-pmap`	Configures an inspect zone type parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood rate per-destination** *maximum-rate* **Example:** `Router(config-profile)# tcp syn-flood rate per-destination 400`	Configures the number of SYN flood packets per second for each destination address. • If the rate of SYN packets sent to a particular destination address exceeds the per-destination limit, the firewall starts processing SYN cookies for SYN packets that are routed to the destination address.
Step 5	**max-destination** *limit* **Example:** `Router(config-profile)# max-destination 10000`	Configures the maximum number of destinations that the firewall can track for a zone. • The firewall drops the SYN packets if the maximum destination crosses the limit that is configured by using the *limit* argument.
Step 6	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode and enters global configuration mode.
Step 7	**zone security** *zone-name* **Example:** `Router(config)# zone security secure-zone`	Configures a security zone and enters security zone configuration mode.
Step 8	**protection** *parameter-map-name* **Example:**	Configures protection for the specified zone using the parameter map.

	Command or Action	Purpose
	`Router(config-sec-zone)# protection zone-pmap`	
Step 9	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration and enters privileged EXEC mode.
Step 10	**show parameter-map type inspect-zone** *zone-pmap-name* **Example:** `Router# show parameter-map type inspect-zone zone-pmap`	(Optional) Displays details about the inspect zone type parameter map.
Step 11	**show zone security** **Example:** `Router# show zone security`	(Optional) Displays zone security information.
Step 12	**show policy-firewall stats zone** *zone-name* **Example:** `Router# show policy-firewall stats zone secure-zone`	(Optional) Displays how many SYN packets exceeded the packet limit and were processed by SYN cookies.

Configuring Firewall Session Table Protection

TCP SYN packets are sent to a range of addresses behind the firewall aiming to exhaust the session table resources on the firewall, thereby denying resources to the legitimate traffic going through the firewall. You can configure firewall session table protection either for the global routing domain or for the VRF domain.

Configuring Firewall Session Table Protection for Global Routing Domain

Perform this task to configure firewall session table protection for global routing domains.

Note A global parameter map takes effect on the global routing domain and not at the router level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **tcp syn-flood limit** *number*
5. **end**
6. **show policy-firewall stats vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect global** **Example:** `Router(config)# parameter-map type inspect global`	Configures a global parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood limit** *number* **Example:** `Router(config-profile)# tcp syn-flood limit 500`	Limits the number of TCP half-open sessions that triggers SYN cookie processing for new SYN packets.
Step 5	**end** **Example:** `Router(config-profile)# end`	Exits profile configuration mode and enters privileged EXEC mode.
Step 6	**show policy-firewall stats vrf global** **Example:** `Router# show policy-firewall stats vrf global`	(Optional) Displays the status of the global VRF firewall policy. • The command output also displays how many TCP half-open sessions are present.

Configuring Firewall Session Table Protection for VRF Domain

Perform this task to configure the firewall session table protection for VRF domains.

Note You can specify the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect-vrf** *vrf-pmap-name*
4. **tcp syn-flood limit** *number*
5. **exit**

6. **parameter-map type inspect global**
7. **vrf** *vrf-name* **inspect** *parameter-map-name*
8. **end**
9. **show parameter-map type inspect-vrf**
10. **show policy-firewall stats vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type inspect-vrf** *vrf-pmap-name* **Example:** `Router(config)# parameter-map type inspect-vrf vrf-pmap`	Configures an inspect-VRF type parameter map and enters profile configuration mode.
Step 4	**tcp syn-flood limit** *number* **Example:** `Router(config-profile)# tcp syn-flood limit 200`	Limits the number of TCP half-open sessions that triggers SYN cookie processing for new SYN packets.
Step 5	**exit** **Example:** `Router(config-profile)# exit`	Exits profile configuration mode and enters global configuration mode.
Step 6	**parameter-map type inspect global** **Example:** `Router(config)# parameter-map type inspect global`	Binds the inspect-VRF type parameter map to a VRF and enters profile configuration mode.
Step 7	**vrf** *vrf-name* **inspect** *parameter-map-name* **Example:** `Router(config-profile)# vrf vrf1 inspect vrf-pmap`	Binds the parameter map to the VRF.
Step 8	**end** **Example:**	Exits profile configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-profile)# end`	
Step 9	**show parameter-map type inspect-vrf** **Example:** `Router# show parameter-map type inspect-vrf`	(Optional) Displays information about inspect VRF type parameter map.
Step 10	**show policy-firewall stats vrf** *vrf-name* **Example:** `Router# show policy-firewall stats vrf vrf-pmap`	(Optional) Displays the status of the VRF firewall policy. • The command output also displays how many TCP half-open sessions are present.

Configuration Examples for Firewall TCP SYN Cookie

Example Configuring Firewall Host Protection

The following example shows how to configure the firewall host protection:

```
Router(config)# parameter-map type inspect-zone zone-pmap
Router(config-profile)# tcp syn-flood rate per-destination 400
Router(config-profile)# max-destination 10000
Router(config-profile)# exit
Router(config)# zone security secure-zone
Router(config-sec-zone)# protection zone-pmap
```

Example Configuring Firewall Session Table Protection

Global Parameter Map

The following example shows how to configure firewall session table protection for global routing domains:

```
Router# configure terminal
Router(config)# parameter-map type inspect global
Router(config-profile)# tcp syn-flood limit 500
```

```
Router(config-profile)# end
```

Inspect-VRF Type Parameter Map

The following example shows how to configure firewall session table protection for VRF domains:

```
Router# configure terminal

Router(config)# parameter-map type inspect-vrf vrf-pmap

Router(config-profile)# tcp syn-flood limit 200

Router(config-profile)# exit

Router(config)# parameter-map type inspect global

Router(config-profile)# vrf vrf1 inspect vrf-pmap

Router(config-profile)# end
```

Additional References for Firewall TCP SYN Cookie

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Firewall TCP SYN Cookie

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 200: Feature Information for Configuring Firewall TCP SYN Cookie

Feature Name	Releases	Feature Information
Firewall TCP SYN Cookie	Cisco IOS XE Release 3.3S	The Firewall TCP SYN Cookie feature protects your firewall from TCP SYN-flooding attacks. TCP SYN-flooding attacks are a type of DoS attack. Usually, TCP SYN packets are sent to a targeted end host or a range of subnet addresses behind the firewall. These TCP SYN packets have spoofed source IP addresses. A spoofing attack is when a person or a program pretends to be another by falsifying data and thereby gaining an illegitimate advantage. The TCP SYN-flooding can take up all the resource on a firewall or an end host, thereby causing DoS to legitimate traffic. To prevent TCP SYN-flooding on a firewall and the end hosts behind the firewall, you must configure the Firewall TCP SYN Cookie feature. The following commands were introduced or modified: **parameter-map type inspect-vrf**, **parameter-map type inspect-zone**, **parameter-map type inspect global**, **show policy-firewall stats**, **tcp syn-flood rate per-destination**, **tcp syn-flood limit**.

CHAPTER 148

Object Groups for ACLs

The Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply these groups to access control lists (ACLs) to create access control policies for these groups. This feature lets you use object groups instead of individual IP addresses, protocols, and ports, which are used in conventional ACLs. This feature allows multiple access control entries (ACEs). You can use each ACE to allow an entire group of users to access a group of servers or services or to deny them access; thereby reducing the size of an ACL and improving manageability.

This module describes object-group ACLs with zone-based policy firewalls and how to configure them for zone-based firewalls.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Object Groups for ACLs

The following restrictions apply to the Object Groups for ACLs feature on zone-based firewalls:

- IPv6 is not supported.
- Dynamic and per-user access control lists (ACLs) are not supported.
- You cannot remove an object group or make an object group empty if it is used in an ACL.

- ACL statements using object groups will be ignored on packets that are sent to RP for processing.
- Object groups are supported only for IP extended ACLs.

Information About Object Groups for ACLs

Overview of Object Groups for ACLs

In large networks, the number of lines in an access control list (ACL) can be large (hundreds of lines) and difficult to configure and manage, especially if the ACLs frequently change. Object group-based ACLs are smaller, more readable, and easier to configure and manage. Object-group-based ACLs simplify static ACL deployments for large user access environments on Cisco IOS routers. The zone-based firewall benefits from object groups, because object groups simplify policy creation (for example, group A has access to group A services).

You can configure conventional access control entries (ACEs) and ACEs that refer to object groups in the same ACL. You can use object-group-based ACLs with quality of service (QoS) match criteria, zone-based policy firewall, Dynamic Host Configuration Protocol (DHCP), and any other features that use extended ACLs.

In addition, you can use object-group-based ACLs with multicast traffic. When there are many inbound and outbound packets, using object group-based ACLs increases performance compared to conventional ACLs. Also, in large configurations, this feature reduces the storage required in NVRAM, because you need not define an individual ACE for every address and protocol pairing.

Integration of Zone-Based Firewalls with Object Groups

Zone-based firewalls use object-group access control lists (ACLs) to apply policies to specific traffic. You define an object-group ACL, associate it with a zone-based firewall policy, and apply the policy to a zone pair to inspect the traffic.

In Cisco IOS XE Release 3.12S, only expanded object-group ACLs are supported with firewalls.

The following features work with object groups that are configured on a firewall:

- Static and dynamic network address translation (NAT)
- Service NAT (NAT that supports non-standard FTP port numbers configured by the **ip nat service** command)
- FTP application layer gateway (ALG)
- Session Initiation Protocol (SIP) ALG

In a class map, you can configure a maximum of 64 matching statements using the **match access-group** command.

Objects Allowed in Network Object Groups

A network object group is a group of any of the following objects:

- IPv6 address
- Host IPv6 addresses

- Other network object groups
- Subnets

Objects Allowed in Service Object Groups

A service object group is a group of any of the following objects:

- Source and destination protocol ports (such as Telnet or Simple Network Management Protocol [SNMP])
- Internet Control Message Protocol (ICMP) types (such as echo, echo-reply, or unreachables)
- Top-level protocols (such as Encapsulating Security Payload [ESP], TCP, or UDP)
- Other service object groups

ACLs Based on Object Groups

All features that use or reference conventional access control lists (ACLs) are compatible with object-group-based ACLs, and the feature interactions for conventional ACLs are the same with object-group-based ACLs. This feature extends the conventional ACLs to support object-group-based ACLs and also adds new keywords and the source and destination addresses and ports.

You can add, delete, or change objects in an object group membership list dynamically (without deleting and redefining the object group). Also, you can add, delete, or change objects in an object group membership list without redefining the ACL access control entry (ACE) that uses the object group. You can add objects to groups, delete them from groups, and then ensure that changes are correctly functioning within the object-group-based ACL without reapplying the ACL to the interface.

You can configure an object-group-based ACL multiple times with a source group only, a destination group only, or both source and destination groups.

You cannot delete an object group that is used within an ACL or a class-based policy language (CPL) policy.

Guidelines for Object Group ACLs

- Object groups must have unique names. For example, to create a network object group named “Engineering” and a service object group named “Engineering,” you must add an identifier (or tag) to at least one object group name to make it unique. For example, you can use the names “Engineering-admins” and “Engineering-hosts” to make the object group names unique and to make it easier for identification.
- Additional objects can be added to an existing object group. After adding an object group, you can add more objects as required for the same group name. You do not need to re-enter existing objects; the previous configuration remains in place until the object group is removed.
- Different objects can be grouped together. For example, objects such as hosts, protocols, or services can be grouped together and configured under the same group name. Network objects can be defined only under a network group, and service objects can be defined only under a service group.
- When you define a group with the **object-group** command and use any security appliance command, the command applies to every item in that group. This feature can significantly reduce your configuration size.

- If an ACL that is associated with a class-map for ZBF inspections includes object-groups, when you add entries to or remove entries from the ACL, the changes take effect only after you exit the access-list configuration prompt.

How to Configure Object Groups for ACLs

To configure object groups for ACLs, you first create one or more object groups. These can be any combination of network object groups (groups that contain objects such as, host addresses and network addresses) or service object groups (which use operators such as **lt**, **eq**, **gt**, **neq**, and **range** with port numbers). Then, you create access control entries (ACEs) that apply a policy (such as **permit** or **deny**) to those object groups.

Creating a Network Object Group

A network object group that contains a single object (such as a single IP address, a hostname, another network object group, or a subnet) or multiple objects with a network object-group-based ACL to create access control policies for the objects.

Perform this task to create a network object group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group network** *object-group-name*
4. **description** *description-text*
5. **host** {*host-address* | *host-name*}
6. *network-address* {*/nn* | *network-mask*}
7. **group-object** *nested-object-group-name*
8. Repeat the steps until you have specified objects on which you want to base your object group.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**object-group network** *object-group-name* **Example:**	Defines the object group name and enters network object-group configuration mode.

	Command or Action	Purpose
	`Device(config)# object-group network my-network-object-group`	
Step 4	**description** *description-text* **Example:** `Device(config-network-group)# description test engineers`	(Optional) Specifies a description of the object group. • You can use up to 200 characters.
Step 5	**host** {*host-address* | *host-name*} **Example:** `Device(config-network-group)# host 209.165.200.237`	(Optional) Specifies the IP address or name of a host. • If you specify a host address, you must use an IPv4 address.
Step 6	*network-address* {*/nn* | *network-mask*} **Example:** `Device(config-network-group)# 209.165.200.225 255.255.255.224`	(Optional) Specifies a subnet object. • You must specify an IPv4 address for the network address. The default network mask is 255.255.255.255.
Step 7	**group-object** *nested-object-group-name* **Example:** `Device(config-network-group)# group-object my-nested-object-group`	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. • The type of child object group must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). • You can use duplicated objects in an object group only via nesting of group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). • You can use an unlimited number of levels of nested object groups (however, a maximum of two levels is recommended).
Step 8	Repeat the steps until you have specified objects on which you want to base your object group.	—
Step 9	**end** **Example:** `Device(config-network-group)# end`	Exits network object-group configuration mode and returns to privileged EXEC mode.

Creating a Service Object Group

Use a service object group to specify TCP and/or UDP ports or port ranges. When the service object group is associated with an access control list (ACL), this service object-group-based ACL can control access to ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **object-group service** *object-group-name*
4. **description** *description-text*
5. *protocol*
6. {**tcp** | **udp** | **tcp-udp**} [**source** {{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*}] [{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*]
7. **icmp** *icmp-type*
8. **group-object** *nested-object-group-name*
9. Repeat the steps to specify the objects on which you want to base your object group.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**object-group service** *object-group-name* **Example:** `Device(config)# object-group service my-service-object-group`	Defines an object group name and enters service object-group configuration mode.
Step 4	**description** *description-text* **Example:** `Device(config-service-group)# description test engineers`	(Optional) Specifies a description of the object group. • You can use up to 200 characters.
Step 5	*protocol* **Example:**	(Optional) Specifies an IP protocol number or name.

	Command or Action	Purpose
	`Device(config-service-group)# ahp`	
Step 6	{**tcp** | **udp** | **tcp-udp**} [**source** {{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*}] [{[**eq**] | **lt** | **gt**} *port1* | **range** *port1 port2*] **Example:** `Device(config-service-group)# tcp-udp range 2000 2005`	(Optional) Specifies TCP, UDP, or both.
Step 7	**icmp** *icmp-type* **Example:** `Device(config-service-group)# icmp conversion-error`	(Optional) Specifies the decimal number or name of an Internet Control Message Protocol (ICMP) type.
Step 8	**group-object** *nested-object-group-name* **Example:** `Device(config-service-group)# group-object my-nested-object-group`	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. • The type of child object group must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). • You can use duplicated objects in an object group only via nesting of group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). • You can use an unlimited number of levels of nested object groups (however, a maximum of two levels is recommended).
Step 9	Repeat the steps to specify the objects on which you want to base your object group.	—
Step 10	**end** **Example:** `Device(config-service-group)# end`	Exits service object-group configuration mode and returns to privileged EXEC mode.

Creating an Object-Group-Based ACL

When creating an object-group-based access control list (ACL), configure an ACL that references one or more object groups. As with conventional ACLs, you can associate the same access policy with one or more interfaces.

You can define multiple access control entries (ACEs) that reference object groups within the same object-group-based ACL. You can also reuse a specific object group in multiple ACEs.

Perform this task to create an object-group-based ACL.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **remark** *remark*
5. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
6. **remark** *remark*
7. **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
8. Repeat the steps to specify the fields and values on which you want to base your access list.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended nomarketing`	Defines an extended IP access list using a name and enters extended access-list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark protect server by denying access from the Marketing network`	(Optional) Adds a comment about the configured access list entry. • A remark can precede or follow an access list entry. • In this example, the remark reminds the network administrator that the subsequent entry denies the Marketing network access to the interface.
Step 5	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence**	(Optional) Denies any packet that matches all conditions specified in the statement.

	Command or Action	Purpose

	precedence] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
	Example:
	`Device(config-ext-nacl)# deny ip 209.165.200.244 255.255.255.224 host 209.165.200.245 log`
	`Example based on object-group:`
	`Router(config)#object-group network my_network_object_group`
	`Router(config-network-group)#209.165.200.224 255.255.255.224`
	`Router(config-network-group)#exit`
	`Router(config)#object-group network my_other_network_object_group`
	`Router(config-network-group)#host 209.165.200.245`
	`Router(config-network-group)#exit`
	`Router(config)#ip access-list extended nomarketing`
	`Router(config-ext-nacl)#deny ip object-group my_network_object_group object-group my_other_network_object_group log`
		• Optionally use the **object-group** *service-object-group-name* keyword and argument as a substitute for the *protocol.* argument
		• Optionally use the **object-group** *source-network-object-group-name* keyword and argument as a substitute for the *source source-wildcard.* arguments
		• Optionally use the **object-group** *destination-network-object-group-name* keyword and argument as a substitute for the *destination destination-wildcard.* arguments
		• If the *source-wildcard* or *destination-wildcard*is omitted, a wildcard mask of 0.0.0.0 is assumed, which matches all bits of the source or destination address, respectively.
		• Optionally use the **any** keyword as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
		• Optionally use the **host** *source* keyword and argument to indicate a source and source wildcard of *source* 0.0.0.0 or the **host** *destination* keyword and argument to indicate a destination and destination wildcard of *destination* 0.0.0.0.
		• In this example, packets from all sources are denied access to the destination network 209.165.200.244. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the **logging facility** command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the **logging console** command.
		•

Step 6	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark allow TCP from any source to any destination`	(Optional) Adds a comment about the configured access list entry. • A remark can precede or follow an access list entry.
Step 7	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence**	Permits any packet that matches all conditions specified in the statement.

	Command or Action	Purpose
	precedence] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	• Every access list needs at least one permit statement. • Optionally use the **object-group** *service-object-group-name* keyword and argument as a substitute for the *protocol.* • Optionally use the **object-group** *source-network-object-group-name* keyword and argument as a substitute for the *source source-wildcard.* • Optionally use the **object-group** *destination-network-object-group-name* keyword and argument as a substitute for the *destination destination-wildcard.* • If *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, which matches on all bits of the source or destination address, respectively. • Optionally use the **any**keyword as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • In this example, TCP packets are allowed from any source to any destination. • Use the **log-input** keyword to include input interface, source MAC address, or virtual circuit in the logging output.
Step 8	Repeat the steps to specify the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 9	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended access-list configuration mode and returns to privileged EXEC mode.

Configuring Class Maps and Policy Maps for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-all** *class-map-name*
4. **match access-group name** *access-list-name*

5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **pass**
9. **exit**
10. **class class-default**
11. **drop**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-all** *class-map-name* **Example:** `Device(config)# class-map type inspect match-all ogacl-cmap`	Creates a Layer 3 and Layer 4 inspect type class map and enters the class-map configuration mode.
Step 4	**match access-group name** *access-list-name* **Example:** `Device(config-cmap)# match access-group name my-ogacl-policy`	Configures a match criterion for a class map on the basis of the specified ACL.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect ogacl-pmap`	Creates a inspect-type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect ogacl-cmap`	Specifies the traffic class on which an action is to be performed and enters policy-map class configuration mode.
Step 8	**pass** **Example:** `Device(config-pmap-c)# pass`	Allows packets to be sent to a device without being inspected.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the default class to configure or modify a policy and enters policy-map class configuration mode.
Step 11	**drop** **Example:** `Device(config-pmap-c)# drop`	Drops packets that are sent to a device.
Step 12	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Configuring Zones for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **interface** *type number*
8. **zone-member security** *zone-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**zone security** *zone-name* **Example:** `Device(config)# zone security outside`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** *zone-name* **Example:** `Device(config)# zone security inside`	Creates a security zone and enters security zone configuration mode. • You need two security zones to create a zone pair: a source zone and a destination zone
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 8	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security inside`	Attaches an interface to a security zone.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Applying Policy Maps to Zone Pairs for Object Groups

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone-pair security** *zone-pair-name* **source** {*zone-name* | **default** | **self**} **destination** {*zone-name* | **default** | **self**}
4. **service-policy type inspect** *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone-pair security** *zone-pair-name* **source** {*zone-name* | **default** | **self**} **destination** {*zone-name* | **default** | **self**} **Example:** `Device(config)# zone-pair security out-to-in source outside destination inside`	Creates a zone pair and enters security zone-pair configuration mode.
Step 4	**service-policy type inspect** *policy-map-name* **Example:** `Device(conf-sec-zone-pair)# service-policy type inspect ogacl-pmap`	Attaches a firewall policy map to a security zone pair.
Step 5	**end** **Example:** `Device(config-sec-zone-pair)# end`	Exits security zone-pair configuration mode and returns to global configuration mode.

Verifying Object Groups for ACLs

SUMMARY STEPS

1. **enable**
2. **show object-group** [*object-group-name*]
3. **show ip access-list** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show object-group** [*object-group-name*] **Example:** `Device# show object-group my-object-group`	Displays the configuration in the named or numbered object group (or in all object groups if no name is entered).

	Command or Action	Purpose
Step 3	**show ip access-list** [*access-list-name*] **Example:** `Device# show ip access-list my-ogacl-policy`	Displays the contents of the named or numbered access list or object group-based ACL (or for all access lists and object group-based ACLs if no name is entered).

Configuration Examples for Object Groups for ACLs

Example: Creating an IPv6 Network Object Group

The following example shows how to create an IPv6 network object group named v6-network ognet1:

```
Device> enable
Device# configure terminal
Device(config)# object-group v6-network ognet1
Device(config-v6-network-group)# 1:1:2::0/32
Device(config-v6-network-group)# host AB:233::23D5
Device(config-v6-network-group)# exit
```

The following example shows how to create a network object group named v6-network ognet2, which contains a host, a subnet, and an existing object group (child) as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group network v6-network ognet2
Device(config-v6network-group)# 1:2:3::4/36
Device(config-v6network-group)# host AABB::CCDD
Device(config-v6network-group)# group-object ognet1
Device(config-v6network-group)# exit
```

Example: Creating a IPv6 Service Object Group

The following example shows how to create a service object group named v6-service ogserv1, which contains several ICMP, TCP, UDP, and TCP-UDP protocols as objects:

```
Device> enable
Device# configure terminal
Device(config)# object-group service v6-service ogserv1
Device(config-v6service-group)# icmp unreachable
Device(config-v6service-group)# tcp smtp
Device(config-v6service-group)# tcp telnet
Device(config-v6service-group)# tcp source range 3000 4000 telnet
Device(config-v6service-group)# pcp
Device(config-v6service-group)# udp domain
Device(config-v6service-group)# hph
Device(config-v6service-group)# exit
```

Example: Creating an IPv6 Object Group-Based ACL

The following example shows how to create an IPv6 object-group-based ACL that permits packets:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list ogacl1
Device(config-ipv6-acl)# permit object-group ogserv1 5:6:7::5/56 object-group ognet1
Device(config-ipv6-acl)# deny ip object-group ognet2 object-group ognet3
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
```

Example: Configuring Class Maps and Policy Maps for Object Groups

```
Device# configure terminal
Device(config)# class-map type inspect match-all ogacl-cmap
Device(config-cmap)# match access-group name my-ogacl-policy
Device(config-cmap)# exit
Device(config)# policy-map type inspect ogacl-pmap
Device(config-pmap)# class type inspect ogacl-cmap
Device(config-pmap-c)# pass
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

Example: Configuring Zones for Object Groups

```
Device# configure terminal
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone-pair security out-to-in source outside destination inside
Device(conf-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# zone-member security outside
Device(config-if)# end
```

Example: Applying Policy Maps to Zone Pairs for Object Groups

```
Device# configure terminal
Device(config)# zone-pair security out-to-in source outside destination inside
Device(config-sec-zone-pair)# service-policy type inspect ogacl-pmap
Device(config-sec-zone-pair)# end
```

Example: Verifying IPv6 Object Groups for ACLs

The following example shows how to display all object groups:

```
Device# show object-group

V6-Network object group ognet1
1:1:2::/32
host AB:233::23D5
V6-Network object group ognet2
1:2:3::4/36
host AABB::CCDD
group-object ognet1
V6-Network object group ognet3
host 1::1
host 1::2
host 1::3
V6-Service object group ogserv1
icmp unreachable
tcp source range 3000 4000 eq telnet
pcp
hbh
```

The following example shows how to display information about IPv6 object-group-based ACL:

```
Device# show ipv6 access-list
IPv6 access list ogacl1
      permit object-group ogserv1 5:6:7::/56 object-group ognet1 sequence 10
      deny ipv6 object-group ognet2 object-group ognet3 sequence 20
      permit ipv6 any any sequence 30
```

Additional References for Object Groups for ACLs

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
ACL configuration guide	*Security Configuration Guide: Access Control Lists*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Object Groups for ACLs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 201: Feature Information for Object Groups for ACLs

Feature Name	Releases	Feature Information
IPv6 Object Groups for ACLs	Cisco IOS XE Release 16.11.1	The IPv6 Object Groups for ACLs feature lets you classify users, devices, or protocols into groups and apply them to access control lists (ACLs) to create access control policies for those groups. This feature lets you use object groups instead of individual IP addresses, protocols, and ports, which are used in conventional ACLs. This feature allows multiple access control entries (ACEs), but now you can use each ACE to allow an entire group of users to access a group of servers or services or to deny them from doing so.

CHAPTER 149

Cisco Firewall-SIP Enhancements ALG

The enhanced Session Initiation Protocol (SIP) inspection in the Cisco XE firewall provides basic SIP inspect functionality (SIP packet inspection and pinholes opening) as well as protocol conformance and application security. These enhancements give you control on what policies and security checks to apply to SIP traffic and the capability to filter out unwanted messages or users.

The development of additional SIP functionality in Cisco IOS XE software provides increased support for Cisco Call Manager, Cisco Call Manager Express, and Cisco IP-IP Gateway based voice/video systems. The application-layer gateway (ALG) SIP enhancement also supports RFC 3261 and its extensions.

Prerequisites for Cisco Firewall-SIP Enhancements ALG

Your system must be running Cisco IOS XE Release 2.4 or a later release.

Restrictions for Cisco Firewall-SIP Enhancements ALG

DNS Name Resolution

Although SIP methods can have Domain Name System (DNS) names instead of raw IP addresses, this feature currently does not support DNS names.

Cisco ASR 1000 Series Routers

This feature was implemented without support for application inspection and control (AIC) on the Cisco ASR 1000 series routers. The Cisco IOS XE Release 2.4 supports the following commands only: **class-map type inspect**, **class type inspect**, **match protocol**, and **policy-map type inspect**.

Cisco ISR 4000 Series Routers

The Cisco IOS XE Fuji 16.7.1 release does not support Transport Layer Security (TLS) or Secure Real-time Transport Protocol (SRTP).

Information About Cisco Firewall-SIP Enhancements ALG

SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations that are used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to users' current locations, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Firewall for SIP Functionality Description

The firewall for SIP support feature allows SIP signaling requests to traverse directly between gateways or through a series of proxies to the destination gateway or phone. After the initial request, if the Record-Route header field is not used, subsequent requests can traverse directly to the destination gateway address as specified in the Contact header field. Thus, the firewall is aware of all surrounding proxies and gateways and allows the following functionalities:

- SIP signaling responses can travel the same path as SIP signaling requests.
- Subsequent signaling requests can travel directly to the endpoint (destination gateway).
- Media endpoints can exchange data between each other.

SIP UDP and TCP Support

RFC 3261 is the current RFC for SIP, which replaces RFC 2543. This feature supports the SIP UDP and the TCP format for signaling.

SIP Inspection

This section describes the deployment scenarios supported by the Cisco Firewall--SIP ALG Enhancements feature.

Cisco IOS XE Firewall Between SIP Phones and CCM

The Cisco IOS XE firewall is located between Cisco Call Manager or Cisco Call Manager Express and SIP phones. SIP phones are registered to Cisco Call Manager or Cisco Call Manager Express through the firewall, and any SIP calls from or to the SIP phones pass through the firewall.

Cisco IOS XE Firewall Between SIP Gateways

The Cisco IOS XE firewall is located between two SIP gateways, which can be Cisco Call Manager, Cisco Call Manager Express, or a SIP proxy. Phones are registered with SIP gateways directly. The firewall sees the SIP session or traffic only when there is a SIP call between phones registered to different SIP gateways. In some scenarios an IP-IP gateway can also be configured on the same device as the firewall. With this scenario all the calls between the SIP gateways are terminated in the IP-IP gateway.

Cisco IOS XE Firewall with Local Cisco Call Manager Express and Remote Cisco Call Manager Express/Cisco Call Manager

The Cisco IOS XE firewall is located between two SIP gateways, which can be Cisco Call Manager, Cisco Call Manager Express, or a SIP proxy. One of the gateways is configured on the same device as the firewall. All the phones registered to this gateway are locally inspected by the firewall. The firewall also inspects SIP sessions between the two gateways when there is a SIP call between them. With this scenario the firewall locally inspects SIP phones on one side and SIP gateways on the other side.

Cisco IOS XE Firewall with Local Cisco Call Manager Express

The Cisco IOS XE firewall and Cisco Call Manager Express is configured on the same device. All the phones registered to the Cisco Call Manager Express are locally inspected by the firewall. Any SIP call between any of the phones registered will also be inspected by the Cisco IOS XE firewall.

ALG--SIP Over TCP Enhancement

When SIP is transferred over UDP, every SIP message is carried in one single UDP datagram. However, when SIP is transferred over TCP, one TCP segment may contain multiple SIP messages. And it is possible that the last SIP message in one of the TCP segments may be a partial one. Prior to Cisco IOS XE Release 3.5S, when there are multiple SIP messages in one received TCP segment, the SIP ALG parses only the first message. The data that is not parsed is regarded as one incomplete SIP message and returned to vTCP. When the next TCP segment is received, vTCP prefixes the unprocessed data to that segment to pass them to the SIP ALG and causes more and more data have to be buffered in vTCP.

In Cisco IOS XE Release 3.5S, the ALG--SIP over TCP Enhancement feature lets the SIP ALG to handle multiple SIP messages in one TCP segment. When a TCP segment is received, all complete SIP messages inside this segment are parsed one-by-one. If there is an incomplete message in the end, only that portion is returned to vTCP.

How to Configure Cisco Firewall-SIP Enhancements ALG

Enabling SIP Inspection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**

6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sip`	Configures the match criterion for a class map based on the named protocol.
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect sip-class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.

	Command or Action	Purpose
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can be used to troubleshoot your SIP-enabled firewall configuration:

- **clear zone-pair**
- **debug cce**
- **debug policy-map type inspect**
- **show policy-map type inspect zone-pair**
- **show zone-pair security**

Configuring a Zone Pair and Attaching a SIP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Device(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and returns to security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sip-policy`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco Firewall-SIP Enhancements ALG

Example: Enabling SIP Inspection

```
class-map type inspect match-any sip-class1
 match protocol sip
!
policy-map type inspect sip-policy
 class type inspect sip-class1
  inspect
!
class class-default
```

Example: Configuring a Zone Pair and Attaching a SIP Policy Map

```
zone security zone1
!
zone security zone2
!
zone-pair security in-out source zone1 destination zone2
 service-policy type inspect sip-policy
!
interface gigabitethernet 0/0/0
 zone security zone1
!
interface gigabitethernet 0/1/1
 zone security zone2
```

Additional References for Cisco Firewall-SIP Enhancements ALG

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Additional SIP Information	Guide to Cisco Systems VoIP Infrastructure Solution for SIP
vTCP support	*vTCP for ALG Support*

Standards and RFCs

Standard/RFC	Title
RFC 3261	SIP: Session Initiation Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco Firewall-SIP Enhancements ALG

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 202: Feature Information for Cisco Firewall-SIP Enhancements: ALG

Feature Name	Releases	Feature Information
AGL--SIP Over TCP Enhancement	Cisco IOS XE Release 3.5S	The ALG--SIP over TCP Enhancement feature lets the SIP ALG to handle multiple SIP messages in one TCP segment. When a TCP segment is received, all complete SIP messages inside this segment are parsed one-by-one. If there is an incomplete message in the end, only that portion is returned to vTCP.
Cisco Firewall--SIP ALG Enhancements	Cisco IOS XE Release 2.4	The Cisco Firewall--SIP ALG Enhancements feature provides voice security enhancements within the firewall feature set in Cisco IOS XE software on the Cisco ASR 1000 series routers. The following commands were implemented without support for Layer 7 (application-specific) syntax, on the Cisco ASR 1000 series routers:**class type inspect, class-map type inspect**, **match protocol**, **policy-map type inspect**.
Firewall--SIP ALG Enhancement for T.38 Fax Relay	Cisco IOS XE Release 2.4.1	The Firewall--SIP ALG Enhancement for T.38 Fax Relay feature provides an enhancement within the Firewall feature set in Cisco IOS XE software on the Cisco ASR 1000 series routers. The feature enables SIP ALG to support T.38 Fax Relay over IP, passing through the firewall on the Cisco ASR 1000 series routers.

CHAPTER 150

MSRPC ALG Support for Firewall and NAT

The MSRPC ALG Support for Firewall and NAT feature provides support for the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). The MSRPC ALG provides deep packet inspection (DPI) of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters to define match criteria that can be searched in an MSRPC packet.

The MSRPC ALG additionally supports the Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco IOS zone-based firewall, Network Address Translation (NAT) and other applications.

Prerequisites for MSRPC ALG Support for Firewall and NAT

- You must enable the Cisco IOS XE firewall and Network Address Translation (NAT) before applying the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on packets.

Note MSRPC ALG is automatically enabled if traffic is sent to TCP port 135 by either Cisco IOS XE firewall or NAT, or both.

Restrictions for MSRPC ALG Support for Firewall and NAT

- Only TCP-based MSRPC is supported.
- You cannot configure the **allow** and **reset** commands together.
- You must configure the **match protocol msrpc** command for DPI.

- Only traffic that reaches destination port 135 is supported. This setting can be changed by configuration.

Information About MSRPC ALG Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

MSRPC

MSRPC is a framework that developers use to publish a set of applications and services for servers and enterprises. RPC is an interprocess communication technique that allows the client and server software to communicate over the network. MSRPC is an application-layer protocol that is used by a wide array of Microsoft applications. MSRPC supports both connection-oriented (CO) and connectionless (CL) Distributed Computing Environment (DCE) RPC modes over a wide variety of transport protocols. All services of MSRPC establish an initial session that is referred to as the primary connection. A secondary session over a port range between 1024 to 65535 as the destination port is established by some services of MSRPC.

For MSRPC to work when firewall and NAT are enabled, in addition to inspecting MSRPC packets, the ALG is required to handle MSRPC specific issues like establishing dynamic firewall sessions and fixing the packet content after the NAT.

By applying MSRPC protocol inspection, most MSRPC services are supported, eliminating the need for Layer 7 policy filters.

MSRPC ALG on Firewall

After you configure the firewall to inspect the MSRPC protocol, the MSRPC ALG starts parsing MSRPC messages. The following table describes the types of Protocol Data Units (PDU) supported by the MSRPC ALG Support on Firewall and NAT feature:

Table 203: Supported PDU Types

PDU	Number	Type	Description
REQUEST	0	call	Initiates a call request.
RESPONSE	2	call	Responds to a call request.
FAULT	3	call	Indicates an RPC runtime, RPC stub, or RPC-specific exception.
BIND	11	association	Initiates the presentation negotiation for the body data.
BIND_ACK	12	association	Accepts a bind request.
BIND_NAK	13	association	Rejects an association request.
ALTER_CONTEXT	14	association	Requests additional presentation negotiation for another interface and/or version, or to negotiate a new security context, or both.
ALTER_CONTEXT_RESP	15	association	Responds to the ALTER_CONTEXT PDU. Valid values are accept or deny.
SHUTDOWN	17	call	Requests a client to terminate the connection and free the related resources.
CO_CANCEL	18	call	Cancels or orphans a connection. This message is sent when a client encounters a cancel fault.
ORPHANED	19	call	Terminates a request that in progress and that has not been entirely transmitted yet, or aborts a (possibly lengthy) response that is in progress.

MSRPC ALG on NAT

When NAT receives an MSRPC packet, it invokes the MSRPC ALG that parses the packet payload and forms a token to translate any embedded IP addresses. This token is passed to NAT, which translates addresses or ports as per your NAT configuration. The translated addresses are then written back into the packet payload by the MSRPC ALG.

If you have configured both the firewall and NAT, NAT calls the ALG first.

MSRPC Stateful Parser

The MSRPC state machine or the parser is the brain of the MSRPC ALG. The MSRPC stateful parser keeps all stateful information within the firewall or NAT depending on which feature invokes the parser first. The parser provides DPI of MSRPC protocol packets. It checks for protocol conformance and detects

out-of-sequence commands and malformed packets. As the packet is parsed, the state machine records various data and fills in the correct token information for NAT and firewall inspection.

How to Configure MSRPC ALG Support for Firewall and NAT

Note By default, MSRPC ALG is automatically enabled when NAT is enabled. There is no need to explicitly enable MSRPC ALG in the NAT-only configuration. You can use the **no ip nat service msrpc** command to disable MSRPC ALG on NAT.

Configuring a Layer 4 MSRPC Class Map and Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any msrpc-cmap`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol msrpc`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS XE stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect msrpc-pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect msrpc-class-map`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.
Step 9	**end** **Example:** `Router(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an MSRPC Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]]
8. **service-policy type inspect** *policy-map-name*

9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Rotuer# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone-name* **Example:** `Router(config)# zone security in-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security out-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]] **Example:** `Router(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits security zone pair configuration mode and enters privileged EXEC mode.

Enabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg vtcp service msrpc**
4. **exit**
5. **set platform hardware qfp active feature alg msrpc tolerance on**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**alg vtcp service msrpc** **Example:** `Rotuer(config)# alg vtcp service msrpc`	Enables vTCP functionality for MSRPC ALG. **Note** By default, MSRPC ALG supports vTCP.
Step 4	**exit** **Example:** `Rotuer(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**set platform hardware qfp active feature alg msrpc tolerance on** **Example:** `Rotuer# set platform hardware qfp active feature alg msrpc tolerance on`	Enables MSRPC unknown message tolerance. **Note** By default, the tolerance is switched off.

Disabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no alg vtcp service msrpc**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no alg vtcp service msrpc** **Example:** `Rotuer(config)# no alg vtcp service msrpc`	Disables vTCP functionality for MSRPC ALG.
Step 4	**end** **Example:** `Rotuer(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for MSRPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 MSRPC Class Map and Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any msrpc-cmap
Router(config-cmap)# match protocol msrpc
Router(config-cmap)# exit
Router(config)# policy-map type inspect msrpc-pmap
Router(config-pmap)# class type inspect msrpc-cmap
Router(config-pmap-c)# inspect
```

```
Router(config-pmap-c)# end
```

Example: Configuring a Zone Pair and Attaching an MSRPC Policy Map

```
Router# configure terminal
Router(config)# zone security in-zone
Router(config-sec-zone)# exit
Router(config)# zone security out-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source in-zone destination out-zone
Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap
Router(config-sec-zone-pair)# end
```

Example: Enabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# alg vtcp service msrpc
Router(config)# end
```

Example: Disabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# no alg vtcp service msrpc
Router(config)# end
```

Feature Information for MSRPC ALG Support for Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 204: Feature Information for MSRPC ALG Support for Firewall and NAT

Feature Name	Releases	Feature Information
MSRPC ALG Support for Firewall and NAT	Cisco IOS XE Release 3.5S	The MSRPC ALG Support for Firewall and NAT feature provides support for the MSRPC ALG on the firewall and NAT. The MSRPC ALG provides deep packet inspection of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters that define match criteria that can be searched in an MSRPC packet. The following commands were introduced or modified: **ip nat service msrpc**, **match protocol msrpc**.
MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT	Cisco IOS XE Release 3.14S	The MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT feature supports Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco firewall, Network Address Translation (NAT) and other applications. The following command was introduced: **alg vtcp service msrpc**.

CHAPTER 151

Sun RPC ALG Support for Firewalls and NAT

The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun Microsystems remote-procedure call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). Sun RPC is an application layer protocol that enables client programs to call functions in a remote server program. This module describes how to configure the Sun RPC ALG.

Restrictions for Sun RPC ALG Support for Firewalls and NAT

- If you configure the inspect action for Layer 4 or Layer 7 class maps, packets that match the Port Mapper Protocol well-known port (111) pass through the firewall without the Layer 7 inspection. Without the Layer 7 inspection, firewall pinholes are not open for traffic flow, and the Sun remote-procedure call (RPC) is blocked by the firewall. As a workaround, configure the **match program-number** command for Sun RPC program numbers.
- Only Port Mapper Protocol Version 2 is supported; none of the other versions are supported.
- Only RPC Version 2 is supported.

Information About Sun RPC ALG Support for Firewalls and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.

- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Sun RPC

The Sun remote-procedure call (RPC) application-level gateway (ALG) performs a deep packet inspection of the Sun RPC protocol. The Sun RPC ALG works with a provisioning system that allows network administrators to configure match filters. Each match filter define a match criterion that is searched in a Sun RPC packet, thereby permitting only packets that match the criterion.

In an RPC, a client program calls procedures in a server program. The RPC library packages the procedure arguments into a network message and sends the message to the server. The server, in turn, uses the RPC library and takes the procedure arguments from the network message and calls the specified server procedure. When the server procedure returns to the RPC, return values are packaged into a network message and sent back to the client.

For a detailed description of the Sun RPC protocol, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Sun RPC ALG Support for Firewalls

You can configure the Sun RPC ALG by using the zone-based firewall that is created by using policies and class maps. A Layer 7 class map allows network administrators to configure match filters. The filters specify the program numbers to be searched for in Sun RPC packets. The Sun RPC Layer 7 policy map is configured as a child policy of the Layer 4 policy map with the **service-policy** command.

When you configure a Sun RPC Layer 4 class map without configuring a Layer 7 firewall policy, the traffic returned by the Sun RPC passes through the firewall, but sessions are not inspected at Layer 7. Because sessions are not inspected, the subsequent RPC call is blocked by the firewall. Configuring a Sun RPC Layer 4 class map and a Layer 7 policy allows Layer 7 inspection. You can configure an empty Layer 7 firewall policy, that is, a policy without any match filters.

Sun RPC ALG Support for NAT

By default, the Sun RPC ALG is automatically enabled when Network Address Translation (NAT) is enabled. You can use the **no ip nat service alg** command to disable the Sun RPC ALG on NAT.

How to Configure Sun RPC ALG Support for Firewalls and NAT

For Sun RPC to work when the firewall and NAT are enabled, the ALG must inspect Sun RPC packets. The ALG also handles Sun RPC-specific issues such as establishing dynamic firewall sessions and fixing the packet content after NAT translation.

Configuring the Firewall for the Sun RPC ALG

You must configure a Layer 7 Sun remote-procedure call (RPC) policy map if you have configured the inspect action for the Sun RPC protocol (that is, if you have specified the **match protocol sunrpc** command in a Layer 4 class map).

We recommend that you do not configure both security zones and inspect rules on the same interface because this configuration may not work.

Perform the following tasks to configure a firewall for the Sun RPC ALG:

Configuring a Layer 4 Class Map for a Firewall Policy

Perform this task to configure a Layer 4 class map for classifying network traffic. When you specify the **match-all** keyword with the **class-map type inspect** command, the Sun RPC traffic matches all Sun remote-procedure call (RPC) Layer 7 filters (specified as program numbers) in the class map. When you specify the **match-any** keyword with the **class-map type inspect** , the Sun RPC traffic must match at least one of the Sun RPC Layer 7 filters (specified as program numbers) in the class map.

To configure a Layer 4 class map, use the **class-map type inspect {match-any | match-all}** *classm-map-name* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol** *protocol-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect** {**match-any**	**match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sunrpc-l4-cmap`	Creates a Layer 4 inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sunrpc`	Configures a match criterion for a class map on the basis of the specified protocol.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map configuration mode and enters privileged EXEC mode.

Configuring a Layer 7 Class Map for a Firewall Policy

Perform this task to configure a Layer 7 class map for classifying network traffic. This configuration enables programs such as mount (100005) and Network File System (NFS) (100003) that use Sun RPC. 100005 and 100003 are Sun RPC program numbers. By default, the Sun RPC ALG blocks all programs.

For more information about Sun RPC programs and program numbers, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Use the **class-map type inspect** *protocol-name* command to configure a Layer 7 class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name*
4. **match program-number** *program-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap`	Creates a Layer 7 (application-specific) inspect type class map and enters QoS class-map configuration mode.
Step 4	**match program-number** *program-number* **Example:** `Device(config-cmap)# match program-number 100005`	Specifies the allowed RPC protocol program number as a match criterion.
Step 5	**end** **Example:**	Exits QoS class-map configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-cmap)# end`	

Configuring a Sun RPC Firewall Policy Map

Perform this task to configure a Sun remote-procedure call (RPC) firewall policy map. Use a policy map to allow packet transfer for each Sun RPC Layer 7 class that is defined in a class map for a Layer 7 firewall policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *protocol-name policy-map-name*
4. **class type inspect** *protocol-name class-map-name*
5. **allow**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap`	Creates a Layer 7 (protocol-specific) inspect type policy map and enters QoS policy-map configuration mode.
Step 4	**class type inspect** *protocol-name class-map-name* **Example:** `Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**allow** **Example:** `Device(config-pmap-c)# allow`	Allows packet transfer.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class** {*class-map-name* | **class-default**}
5. **inspect** [*parameter-map-name*]
6. **service-policy** *protocol-name policy-map-name*
7. **exit**
8. **class class-default**
9. **drop**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sunrpc-l4-pmap`	Creates a Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 4	**class** {*class-map-name*	**class-default**} **Example:** `Device(config-pmap)# class sunrpc-l4-cmap`	Associates (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 6	**service-policy** *protocol-name policy-map-name* **Example:** `Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap`	Attaches the Layer 7 policy map to a top-level Layer 4 policy map.
Step 7	**exit** **Example:**	Exits QoS policy-map class configuration mode and returns to QoS policy-map configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c)# exit`	
Step 8	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the default class (commonly known as the class-default class) before you configure its policy and enters QoS policy-map class configuration mode.
Step 9	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 10	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

You need two security zones to create a zone pair. However, you can create only one security zone and the second one can be the system-defined security zone. To create the system-defined security zone or self zone, configure the **zone-pair security** command with the **self** keyword.

Note If you select a self zone, you cannot configure the inspect action.

In this task, you will do the following:

- Create security zones.
- Define zone pairs.
- Assign interfaces to security zones.
- Attach a policy map to a zone pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
12. **zone-member security** *zone-name*

13. **exit**
14. **interface** *type number*
15. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
16. **zone-member security** *zone-name*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-client`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone or self zone as either the source or destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-server`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone as either the source or destination zone.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:**	Creates a zone pair and enters security zone-pair configuration mode.

	Command or Action	Purpose
	`Device(config)# zone-pair security clt2srv source z-client destination z-server`	
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap`	Attaches a firewall policy map to a zone pair.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/0/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 12	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-client`	Attaches an interface to a security zone.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/1/1`	Configures an interface type and enters interface configuration mode.
Step 15	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 16	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-server`	Attaches an interface to a security zone.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Sun RPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect match-any sunrpc-l4-cmap
Device(config-cmap)# match protocol sunrpc
Device(config-cmap)# end
```

Example: Configuring a Layer 7 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap
Device(config-cmap)# match program-number 100005
Device(config-cmap)# end
```

Example: Configuring a Sun RPC Firewall Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap
Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap
Device(config-pmap-c)# allow
Device(config-pmap-c)# end
```

Example: Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpcl4-pmap
Device(config-pmap)# class sunrpcl4-cmap
Device(config-pmap-c)# inspect
Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

Example: Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

```
Device# configure terminal
Device(config)# zone security z-client
Device(config-sec-zone)# exit
```

```
Device(config)# zone security z-server
Device(config-sec-zone)# exit
Device(config)# zone-pair security clt2srv source z-client destination z-server
Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 2/0/0
Device(config-if)# ip address 192.168.6.5 255.255.255.0
Device(config-if)# zone-member security z-client
Device(config-if)# exit
Device(config)# interface gigabitethernet 2/1/1
Device(config-if)# ip address 192.168.6.1 255.255.255.0
Device(config-if)# zone-member security z-server
Device(config-if)# end
```

Example: Configuring the Firewall for the Sun RPC ALG

The following is a sample firewall configuration for the Sun remote-procedure call (RPC) application-level gateway (ALG) support:

```
class-map type inspect sunrpc match-any sunrpc-l7-cmap
 match program-number 100005
!
class-map type inspect match-any sunrpc-l4-cmap
 match protocol sunrpc
!
!
policy-map type inspect sunrpc sunrpc-l7-pmap
 class type inspect sunrpc sunrpc-l7-cmap
  allow
!
!
policy-map type inspect sunrpc-l4-pmap
 class type inspect sunrpc-l4-cmap
  inspect
  service-policy sunrpc sunrpc-l7-pmap
!
 class class-default
  drop
!
!
zone security z-client
!
zone security z-server
!
zone-pair security clt2srv source z-client destination z-server
 service-policy type inspect sunrpc-l4-pmap
!
interface GigabitEthernet 2/0/0
 ip address 192.168.10.1 255.255.255.0
 zone-member security z-client
!
interface GigabitEthernet 2/1/1
 ip address 192.168.23.1 255.255.255.0
 zone-member security z-server
!
```

Additional References for Sun RPC ALG Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
IP Addressing commands	IP Addressing Services Command Reference
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 1057	*RPC: Remote Procedure Call Protocol Specification Version 2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Sun RPC ALG Support for Firewalls and NAT

Table 205: Feature Information for Sun RPC ALG Support for Firewalls and NAT

Feature Name	Releases	Feature Information
Sun RPC ALG Support for Firewalls and NAT	Cisco IOS XE Release 3.2S	The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun RPC ALG on the firewall and NAT. The following command was introduced or modified: **match protocol**.

CHAPTER 152

Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

The Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing feature supports the following functionalities for Application Layer Gateway (ALG), and Application Inspection and Control (AIC):

- Packet tracing
- Conditional debugging
- Debug logs

Information About Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

Packet Tracing

Packet tracing provides the ability to generate Control Plane Policing (CPP) statistics for a specified packet flow, with minimal effect on router throughput. It also traces the path of each packet in the flow, which helps in determining the input interface, features used, and the output path.

Application layer gateway (ALG) generates statistics and keeps a log of the path along which the packets travel.

Conditional Debugging

In a typical Application layer gateway (ALG)-enabled scenario where certain connections from the source address or destination address fail, debugging displays a list of messages for all the traffic that passes through

the ALG. Enabling conditional debugging ensures that debug messages related to specified connections are displayed on the console. Prior to the introduction of this feature, debugging used to display many messages for all traffic that passes through the ALG.

Debug Logs

The following severity levels have been added:

1. Error: Error and firewall packet drop conditions.

   Examples:

   - Unable to send a packet
   - ALG error condition

2. Warning: Warning debug messages.
3. Info: Information about an event.

   Examples:

   - Packet drop due to policy configuration, malformed packets, or hardcoded limit and threshold
   - State machine transition
   - ALG check status
   - Packet pass and drop status

4. Verbose: All log messages.

   Examples:

   - Data structures
   - Event details

Note Both the ALG-AIC functional debug flag and the severity level must be set. If only the severity level is set and the ALG-AIC functional debug flag is not set, the debug log will not be enabled. If only the ALG-AIC functional debug flag is set, the Info level, which is the default severity level, is logged.

Additional References for Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 206: Feature Information for Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support

Feature Name	Releases	Feature Information
Zone-Based Firewall ALG and AIC Conditional Debugging and Packet Tracing Support	Cisco IOS XE 3.13S	The feature supports the following functionalities: • Packet tracing • Conditional debugging • Debug logs

CHAPTER 153

ALG—H.323 vTCP with High Availability Support for Firewall and NAT

The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. Virtual TCP (vTCP) supports TCP segment reassembly. Prior to this introduction of the feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.

This module describes how to configure the ALG—H.323 vTCP with high availability (HA) support for firewalls.

Restrictions for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

- When an incoming TCP segment is not a complete H.323 message, the H.323 ALG buffers the TCP segment while waiting for the rest of the message. The buffered data is not synchronized to the standby device for high availability (HA).
- The performance of the H.323 ALG may get impacted when vTCP starts to buffer data.

Information About ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Basic H.323 ALG Support

H.323 is a recommendation published by the ITU-T defining a series of network elements and protocols for multimedia transmission through packet-based networks. H.323 defines a number of network elements used in multimedia transmission.

Although most H.323 implementations today utilize TCP as the transport mechanism for signaling, H.323 Version 2 enables basic UDP transport.

- H.323 Terminal—This element is an endpoint in the network, providing two-way communication with another H.323 terminal or gateway.
- H.323 Gateway—This element provides protocol conversion between H.323 terminals and other terminals that do not support H.323.
- H.323 Gatekeeper—This element provides services like address translation, network access control, and bandwidth management and account for H.323 terminals and gateways.

The following core protocols are described by the H.323 specification:

- H.225—This protocol describes call signaling methods used between any two H.323 entities to establish communication.
- H.225 Registration, Admission, and Status (RAS)—This protocol is used by the H.323 endpoint and gateway for address resolution and admission control services.
- H.245—This protocol is used for exchanging the capabilities of multimedia communication and for the opening and closing of logical channels for audio, video, and data.

In addition to the protocols listed, the H.323 specification describes the use of various IETF protocols like the Real Time Transport (RTP) protocol and audio (G.711, G.729, and so on) and video (H.261, H.263, and H.264) codecs.

NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The H.323 ALG performs these specific services for H.323 messages.

Overview of vTCP for ALG Support

When a Layer 7 protocol uses TCP for transportation, the TCP payload can be segmented due to various reasons, such as application design, maximum segment size (MSS), TCP window size, and so on. The application-level gateways (ALGs) that the firewall and NAT support do not have the capability to recognize TCP fragments for packet inspection. vTCP is a general framework that ALGs use to understand TCP segments and to parse the TCP payload.

vTCP helps applications like NAT and Session Initiation Protocol (SIP) that require the entire TCP payload to rewrite the embedded data. The firewall uses vTCP to help ALGs support data splitting between packets.

When you configure firewall or NAT ALGs, the vTCP functionality is activated.

vTCP currently supports Real Time Streaming Protocol (RTSP) and DNS ALGs.

TCP Acknowledgment and Reliable Transmission

Because vTCP resides between two TCP hosts, a buffer space is required to store TCP segments temporarily, before they are sent to other hosts. vTCP ensures that data transmission occurs properly between hosts. vTCP sends a TCP acknowledgment (ACK) to the sending host if vTCP requires more data for data transmission. vTCP also keeps track of the ACKs sent by the receiving host from the beginning of the TCP flow to closely monitor the acknowledged data.

vTCP reassembles TCP segments. The IP header and the TCP header information of the incoming segments are saved in the vTCP buffer for reliable transmission.

vTCP can make minor changes in the length of outgoing segments for NAT-enabled applications. vTCP can either squeeze the additional length of data to the last segment or create a new segment to carry the extra data. The IP header or the TCP header content of the newly created segment is derived from the original incoming segment. The total length of the IP header and the TCP header sequence numbers are adjusted accordingly.

vTCP with NAT and Firewall ALGs

ALG is a subcomponent of NAT and the firewall. Both NAT and the firewall have a framework to dynamically couple their ALGs. When the firewall performs a Layer 7 inspection or NAT performs a Layer 7 fix-up, the parser function registered by the ALGs is called and ALGs take over the packet inspection. vTCP mediates between NAT and the firewall and the ALGs that use these applications. In other words, packets are first processed by vTCP and then passed on to ALGs. vTCP reassembles the TCP segments in both directions within a TCP connection.

Overview of ALG—H.323 vTCP with High Availability Support

The ALG-H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. After the H.323 ALG is coupled with vTCP, the firewall and NAT interact with the H.323 ALG through vTCP. When

vTCP starts to buffer data, the high availability (HA) function is impacted, because vTCP cannot synchronize the buffered data to a standby device. If the switchover to the standby device happens when vTCP is buffering data, the connection may be reset if the buffered data is not synchronized to the standby device. After the buffered data is acknowledged by vTCP, the data is lost and the connection is reset. The firewall and NAT synchronize the data for HA. vTCP only synchronizes the status of the current connection to the standby device, and in case of errors, the connection is reset.

How to Configure ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Configuring ALG—H.323 vTCP with High Availability Support for Firewalls

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **exit**
11. **class class-default**
12. **exit**
13. **zone security** *zone-name*
14. **exit**
15. **zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone*
16. **service-policy type inspect** *policy-map-name*
17. **exit**
18. **interface** *type number*
19. **zone member security** *zone-name*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any h.323-class`	Creates an inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol h323`	Configures the match criteria for a class map on the basis of the named protocol.
Step 5	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol h323ras`	Configures the match criteria for a class map on the basis of the named protocol.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect h.323-policy`	Creates an inspect type policy map and enters QoS policy-map configuration mode.
Step 8	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect h.323-class`	Specifies the class on which the action is performed and enters QoS policy-map class configuration mode.
Step 9	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 10	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits QoS policy-map class configuration mode and enters policy-map configuration mode.
Step 11	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Applies the policy map settings to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 13	**zone security** *zone-name* **Example:** `Device(config)# zone security inside`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source and a destination zone. • In a zone pair, you can use the default zone as either the source or the destination zone.
Step 14	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 15	**zone-pair security** *zone-pair-name* **source** *source-zone* **destination** *destination-zone* **Example:** `Device(config)# zone-pair security inside-outside source inside destination outside`	Creates a pair of security zones and enters security-zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 16	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect h.323-policy`	Attaches a firewall policy map to the destination zone pair. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 17	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 18	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 19	**zone member security** *zone-name* **Example:** `Device(config-if)# zone member security inside`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 20	**end** **Example:**	Exits interface configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Configuration Examples for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Example: Configuring ALG—H.323 vTCP with High Availability Support for Firewalls

```
Device# configure terminal
Device(config)# class-map type inspect h.323-class
Device(config-cmap)# match protocol h323
Device(config-cmap)# match protocol h323ras
Device(config-cmap)# exit
Device(config)# policy-map type inspect h323-policy
Device(config-pmap)# class type inspect h323
Device(config-pmap-c)# inspect
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap)# exit
Device(config)# zone security inside
Device(config-sec-zone)# exit
Device(config)# zone security outside
Device(config-sec-zone)# exit
Device(config)# zone-pair security inside-outside source inside destination outside
Device(config-sec-zone-pair)# service-policy type inspect h.323-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# zone-member security inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# zone-member security outside
Device(config-if)# end
```

Additional References for ALG-H.323 vTCP with High Availability Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases

Related Topic	Document Title
Firewall commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
NAT commands	IP Addressing Services Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 207: Feature Information for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Feature Name	Releases	Feature Information
ALG—H.323 vTCP with High Availability Support for Firewall and NAT	Cisco IOS XE Release 3.7S	The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 ALG to support a TCP segment that is not a single H.323 message. vTCP supports segment reassembly. Prior to the introduction of this feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.

CHAPTER 154

SIP ALG Hardening for NAT and Firewall

The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing Session Initiation Protocol (SIP) application-level gateway (ALG) support for Network Address Translation (NAT) and firewall. This feature provides the following enhancements:

- Management of the local database for all SIP Layer 7 data
- Processing of the Via header
- Support for logging additional SIP methods
- Support for Provisional Response Acknowledgment (PRACK) call flow
- Support for the Record-Route header

The above enhancements are available by default; no additional configuration is required on NAT or firewall.

This module explains the SIP ALG enhancements and describes how to enable NAT and firewall support for SIP.

Restrictions for SIP ALG Hardening for NAT and Firewall

- Session Initiation Protocol (SIP) application-level gateway (ALG) does not provide any security features.
- SIP ALG manages the local database based on call IDs. There might be a corner case involving two calls coming from two different clients with the same call ID, resulting in call ID duplication.

Information About SIP ALG Hardening for NAT and Firewall

SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations that are used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to users' current locations, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SIP ALG Local Database Management

A Session Initiation Protocol (SIP) trunk is a direct connection of an IP PBX to a service provider over an IP network using SIP. There can be numerous concurrent calls in a SIP trunk. During the call setup process, all calls use the same control channel for call establishment. More than one call uses the same control channel for call setup. When the same control channel is used by more than one call, the stateful information stored in the control-channel sessions becomes unreliable. SIP stateful information consists of media channel information such as the IP address and port number used by client and server endpoints to send media data. The media channel information is used to create a firewall pinhole and a Network Address Translation (NAT) door for the data channel in firewall and NAT, respectively. Because multiple calls use the same control channel for call setup, there will be multiple sets of media data.

In a SIP trunk, more than one call shares the same firewall and NAT session. NAT and firewall identify and manage a SIP session by using the 5 tuple in a SIP packet—source address, destination address, source port,

destination port, and protocol. The conventional method of using the 5 tuple to identify and match calls does not completely support SIP trunking and often leads to Layer 7 data memory leaks and call matching issues.

In contrast to other application-level gateways (ALGs), SIP ALG manages the SIP Layer 7 data by using a local database to store all media-related information contained in normal SIP calls and in SIP calls embedded in a SIP trunk. SIP ALG uses the Call-ID header field contained in a SIP message to search the local database for call matching and to manage and terminate calls. The Call-ID header field is a dialog identifier that identifies messages belonging to the same SIP dialog.

SIP ALG uses the call ID to perform search in the local database and to manage memory resources. In certain scenarios where SIP ALG is unable to free up a Layer 7 data record from the database, a session timer is used to manage and free resources to ensure that there are no stalled call records in the database.

Note Because all Layer 7 data is managed by SIP ALG by using a local database, SIP ALG never replies on firewall and NAT to free SIP Layer 7 data; SIP ALG frees the data by itself. If you use the **clear** command to clear all NAT translations and firewall sessions, the SIP Layer 7 data in the local database is not freed.

SIP ALG Via Header Support

A Session Initiation Protocol (SIP) INVITE request contains a Via header field. The Via header field indicates the transport paths taken by a SIP request. The Via header also contains information about the return path for subsequent SIP responses, which includes the IP address and the port to which the response message is to be sent.

SIP ALG creates a firewall pinhole or a Network Address Translation (NAT) door based on the first value in the Via header field for each SIP request received, except the acknowledge (ACK) message. If the port number information is missing from the first Via header, the port number is assumed to be 5060.

SIP ALG Method Logging Support

The SIP ALG Hardening for NAT and Firewall feature provides support for detailed logging of the following methods in Session Initiation Protocol (SIP) application-level gateway (ALG) statistics:

- PUBLISH
- OPTIONS
- 1XX (excluding 100,180,183)
- 2XX (excluding 200)

The existing SIP methods that are logged in SIP ALG statistics include ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, REFER, REGISTER, SUBSCRIBE, and 1XX-6XX.

SIP ALG PRACK Call-Flow Support

Session Initiation Protocol (SIP) defines two types of responses: final and provisional. Final responses convey the result of processing a request and are sent reliably. Provisional responses, on the other hand, provide information about the progress of processing a request but are not sent reliably.

Provisional Response Acknowledgement (PRACK) is a SIP method that provides an acknowledgment (ACK) system for provisional responses. PRACK allows reliable exchanges of SIP provisional responses between SIP endpoints. SIP reliable provisional responses ensure that media information is exchanged and resource reservation can occur before connecting the call.

SIP uses the connection, media, and attribute fields of the Session Description Protocol (SDP) during connection negotiation. SIP application-level gateway (ALG) supports SDP information within a PRACK message. If media information exists in a PRACK message, SIP ALG retrieves and processes the media information. SIP ALG also handles the creation of media channels for subsequent media streams. SIP ALG creates a firewall pinhole and a NAT door based on the SDP information in PRACK messages.

SIP ALG Record-Route Header Support

The Record-Route header field is added by a Session Initiation Protocol (SIP) proxy to a SIP request to force future requests in a SIP dialog to be routed through the proxy. Messages sent within a dialog then traverse all SIP proxies, which add a Record-Route header field to the SIP request. The Record-Route header field contains a globally reachable Uniform Resource Identifier (URI) that identifies the proxy.

SIP application-level gateway (ALG) parses the Contact header and uses the IP address and the port value in the Contact header to create a firewall pinhole and a Network Address Translation (NAT) door. In addition, SIP ALG supports the parsing of the Record-Route header to create a firewall pinhole and a NAT door for future messages that are routed through proxies.

How to Configure SIP ALG Hardening for NAT and Firewall

Enabling NAT for SIP Support

NAT support for SIP is enabled by default on port 5060. If this feature has been disabled, perform this task to re-enable NAT support for SIP. To disable the NAT support for SIP, use the **no ip nat service sip** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service sip** {**tcp** | **udp**} **port** *port-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat service sip** {**tcp** | **udp**} **port** *port-number* **Example:** `Device(config)# ip nat service sip tcp port 5060`	Enables NAT support for SIP.
Step 4	**end** **Example:** `Device(config)# end`	Exist global configuration mode and returns to privileged EXEC mode.

Enabling SIP Inspection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sip`	Configures the match criterion for a class map based on the named protocol.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect sip-class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Zone Pair and Attaching a SIP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*

9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Device(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and returns to security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sip-policy`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for SIP ALG Hardening for NAT and Firewall

Example: Enabling NAT for SIP Support

```
Device> enable
Device# configure terminal
Device(config)# ip nat service sip tcp port 5060
Device(config)# end
```

Example: Enabling SIP Inspection

```
class-map type inspect match-any sip-class1
 match protocol sip
!
policy-map type inspect sip-policy
 class type inspect sip-class1
  inspect
!
class class-default
```

Example: Configuring a Zone Pair and Attaching a SIP Policy Map

```
zone security zone1
!
zone security zone2
!
zone-pair security in-out source zone1 destination zone2
 service-policy type inspect sip-policy
!
interface gigabitethernet 0/0/0
 zone security zone1
!
interface gigabitethernet 0/1/1
 zone security zone2
```

Additional References for SIP ALG Hardening for NAT and Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT configuration	*IP Addressing: NAT Configuration Guide*

Related Topic	Document Title
Firewall configuration	*Security Configuration Guide: Zone-Based Policy Firewall*
NAT commands	Cisco IOS IP Addressing Services Command Reference
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 3261	*SIP: Session Initiation Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SIP ALG Hardening for NAT and Firewall

Table 208: Feature Information for SIP ALG Hardening for NAT and Firewall

Feature Name	Releases	Feature Information
SIP ALG Hardening for NAT and Firewall	Cisco IOS XE Release 3.8S	The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing SIP ALG support for NAT and firewall.

CHAPTER 155

SIP ALG Resilience to DoS Attacks

The SIP ALG Resilience to DoS Attacks feature provides protection against Session Initiation Protocol (SIP) application layer gateway (ALG) denial of service (DoS) attacks. This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks.

This module explains the feature and how to configure DoS prevention for the SIP application layer gateway (ALG). Network Address Translation and zone-based policy firewalls support this feature.

Information About SIP ALG Resilience to DoS Attacks

SIP ALG Resilience to DoS Attacks Overview

The SIP ALG Resilience to DoS Attacks feature provides protection against denial of service (DoS) attacks to the Session Initiation Protocol (SIP) application layer gateway (ALG). This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks. This feature is supported by Network Address Translation (NAT) and zone-based policy firewalls.

SIP is an application-level signaling protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP DoS attacks are a major threat to networks.

The following are types of SIP DoS attacks:

- SIP register flooding: A registration flood occurs when many VoIP devices try to simultaneously register to a network. If the volume of registration messages exceeds the device capability, some messages are lost. These devices then attempt to register again, adding more congestion. Because of the network congestion, users may be unable to access the network for some time.
- SIP INVITE flooding: An INVITE flood occurs when many INVITE messages are sent to servers that cannot support all these messages. If the attack rate is very high, the memory of the server is exhausted.
- SIP broken authentication and session attack: This attack occurs when an attacker presumes the identity of a valid user, using digest authentication. When the authentication server tries to verify the identity of the attacker, the verification is ignored and the attacker starts a new request with another session identity. These attacks consume the memory of the server.

SIP ALG Dynamic Blacklist

One of the common methods of denial of service (DoS) attacks involves saturating the target network with external communication requests making the network unable to respond to legitimate traffic. To solve this issue, the SIP ALG Resilience to DoS Attacks feature uses configurable blocked lists. A blocked list is a list of entities that are denied a particular privilege, service, or access. Dynamic blacklists are disabled by default. When requests to a destination address exceed a predefined trigger criteria in the configured blocked list, the Session Initiation Protocol (SIP) application layer gateway (ALG) will drop these packets.

The following abnormal SIP session patterns are monitored by dynamic blocked lists:

- In the configured period of time if a source sends multiple requests to a destination and receives non-2xx (as per RFC 3261, any response with a status code between 200 and 299 is a "2xx response") final responses from the destination.
- In the configured period of time if a source sends multiple requests to a destination and does not receive any response from the destination.

SIP ALG Lock Limit

Both Network Address Translation (NAT) and the firewall use the Session Initiation Protocol (SIP) application layer gateway (ALG) to parse SIP messages and create sessions through tokens. To maintain session states, the SIP ALG uses a per call data structure and Layer 7 data to store call-related information that is allocated when a session is initiated and freed when a session is released. If the SIP ALG does not receive a message that indicates that the call has ended, network resources are held for the call.

Because Layer 7 data is shared between threads, a lock is required to access the data. During denial of service (DoS) and distributed DoS attacks, many threads wait to get the same lock, resulting in heavy CPU usage, which makes the system unstable. To prevent the system from becoming unstable, a limit is added to restrict the number of threads that can wait for a lock. SIP sessions are established by request/response mode. When there are too many concurrent SIP messages for one SIP call, packets that exceed the lock limit are dropped.

SIP ALG Timers

To exhaust resources on Session Initiation Protocol (SIP) servers, some denial of service (DoS) attacks do not indicate the end of SIP calls. To prevent these types of DoS attacks, a protection timer is added.

The SIP ALG Resilience to DoS Attacks feature uses the following timers:

- Call-duration timer that controls the maximum length of an answered SIP call.
- Call-proceeding timer that controls the maximum length of an unanswered SIP call.

When the configured maximum time is reached, the SIP application layer gateway (ALG) releases resources for this call, and future messages related to this call may not be properly parsed by the SIP ALG.

How to Configure SIP ALG Resilience to DoS Attacks

Configuring SIP ALG Resilience to DoS Attacks

You can configure the prevention of denial of service (DoS) parameters for the Session Initiation Protocol (SIP) application layer gateway (ALG) that is used by Network Address Translation (NAT) and the zone-based policy firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg sip processor session max-backlog** *concurrent-processor-usage*
4. **alg sip processor global max-backlog** *concurrent-processor-usage*
5. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address*
6. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*]
7. **alg sip timer call-proceeding-timeout** *time*
8. **alg sip timer max-call-duration** *seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**alg sip processor session max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor session max-backlog 5`	Sets a per session limit for the number of backlog messages waiting for shared resources.
Step 4	**alg sip processor global max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor global max-backlog 5`	Sets the maximum number of backlog messages waiting for shared resources for all SIP sessions.

	Command or Action	Purpose
Step 5	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address* **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1`	Configures dynamic SIP ALG blacklist criteria for the specified destination IP address.
Step 6	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*] **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30`	Configures the time period, in seconds, when packets from a source are blocked if the configured limit is exceeded.
Step 7	**alg sip timer call-proceeding-timeout** *time* **Example:** `Device(config)# alg sip timer call-proceeding-timeout 35`	Sets the maximum time interval, in seconds, to end SIP calls that do not receive a response.
Step 8	**alg sip timer max-call-duration** *seconds* **Example:** `Device(config)# alg sip timer max-call-duration 90`	Sets the maximum call duration, in seconds, for a successful SIP call.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying SIP ALG Resilience to DoS Attacks

Use the following commands to troubleshoot the feature.

SUMMARY STEPS

1. **enable**
2. **show alg sip**
3. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**
4. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**
5. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**
6. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**
7. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**
8. **debug alg** {**all** | **info** | **trace** | **warn**}

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show alg sip**

Displays all Session Initiation Protocol (SIP) application layer gateway (ALG) information.

Example:

```
Device# show alg sip

sip timer configuration
   Type                     Seconds
   max-call-duration        380
   call-proceeding-timeout  620

sip processor configuration
   Type           Backlog number
   session         14
   global          189

sip blacklist configuration
   dst-addr         trig-period(ms)    trig-size     block-time(sec)
   10.0.0.0             60                 30              2000
   10.1.1.1             20                 30              30
   192.0.2.115          1000               5               30
   198.51.100.34        20                 30              388
```

Step 3 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**

Displays SIP ALG-specific statistics information in the Cisco Quantum Flow Processor (QFP).

Example:

```
Device# show platform hardware qfp active feature alg statistics sip

Events
...
Cr dbl entry:                10   Del dbl entry:                10
Cr dbl cfg entry:             8   Del dbl cfg entry:             4
start dbl trig tmr:          10   restart dbl trig tmr:       1014
stop dbl trig tmr:           10   dbl trig timeout:           1014
start dbl blk tmr:            0   restart dbl blk tmr:           0
stop dbl blk tmr:             0   dbl blk tmr timeout:           0
start dbl idle tmr:          10   restart dbl idle tmr:        361
stop dbl idle tmr:            1   dbl idle tmr timeout:          9

DoS Errors
Dbl Retmem Failed:            0    Dbl Malloc Failed:            0
DblCfg Retm Failed:           0    DblCfg Malloc Failed:         0
Session wlock ovflw:          0    Global wlock ovflw:           0
Blacklisted:                561
```

Step 4 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**

Displays brief information about all SIP blocked list data.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dbl
```

```
SIP dbl pool used chunk entries number: 1

 entry_id          src_addr         dst_addr         remaining_time(sec)
 a4a051e0a4a1ebd   10.74.30.189     10.74.5.30               25
```

Step 5 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**

Displays all SIP blocked list settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dblcfg

SIP dbl cfg pool used chunk entries number: 4
 dst_addr        trig_period(ms)    trig_size    block_time(sec)
 10.1.1.1              20              30             30
 10.74.5.30            1000             5             30
 192.0.2.2             60              30           2000
 198.51.100.115        20              30            388
```

Step 6 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**

Displays SIP processor settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip processor

Session:     14        Global:    189

Current global wlock count:      0
```

Step 7 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**

Displays SIP timer settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip timer

call-proceeding:    620      call-duration:    380
```

Step 8 **debug alg** {**all** | **info** | **trace** | **warn**}

Example:

```
Device# debug alg warn
```

Enables the logging of ALG warning messages.

Configuration Examples for SIP ALG Resilience to DoS Attacks

Example: Configuring SIP ALG Resilience to DoS Attacks

```
Device# configure terminal
Device(config)# alg sip processor session max-backlog 5
Device(config)# alg sip processor global max-backlog 5
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30
Device(config)# alg sip timer call-proceeding-timeout 35
Device(config)# alg sip timer max-call-duration 90
Device(config)# end
```

Additional References for SIP ALG Resilience to DoS Attacks

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
NAT commands	IP Addressing Services Command References

Standards and RFCs

Standard/RFC	Title
RFC 4028	*Session Timers in the Session Initiation Protocol (SIP)*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

PART XIII

Security for VPNs with IPsec

CHAPTER 156

Configuring Security for VPNs with IPsec

This module describes how to configure basic IPsec VPNs. IPsec is a framework of open standards developed by the IETF. It provides security for the transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (“peers”), such as Cisco routers.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Configuring Security for VPNs with IPsec

IKE Configuration

You must configure Internet Key Exchange (IKE) as described in the module *Configuring Internet Key Exchange for IPsec VPNs*.

Note If you decide not to use IKE, you must still disable it as described in the module *Configuring Internet Key Exchange for IPsec VPNs*.

Ensure Access Lists Are Compatible with IPsec

IKE uses UDP port 500. The IPsec encapsulating security payload (ESP) and authentication header (AH) protocols use protocol numbers 50 and 51, respectively. Ensure that your access lists are configured so that traffic from protocol 50, 51, and UDP port 500 are not blocked at interfaces used by IPsec. In some cases, you might need to add a statement to your access lists to explicitly permit this traffic.

Restrictions for Configuring Security for VPNs with IPsec

Cisco IPsec Policy Map MIB

The MIB OID objects are displayed only when an IPsec session is up.

Discontiguous Access Control Lists

Crypto maps using access control lists (ACLs) that have discontiguous masks are not supported.

Physical Interface and Crypto Map

A crypto map on a physical interface is not supported, if the physical interface is the source interface of a tunnel protection interface.

NAT Configuration

If you use Network Address Translation (NAT), you should configure static NAT so that IPsec works properly. In general, NAT should occur before the router performs IPsec encapsulation; in other words, IPsec should work with global addresses.

Unicast IP Datagram Application Only

IPsec can be applied to unicast IP datagrams only. Because the IPsec Working Group has not yet addressed the issue of group key distribution, IPsec does not currently work with multicasts or broadcast IP datagrams.

Unsupported Interface Types

- Crypto VPNs are not supported on the bridge domain interfaces (BDI).
- Crypto maps are not supported on tunnel interface and port-channel interface. As an exception, crypto maps for GDOI are supported on tunnel interfaces.
- Crypto maps are not supported on loopback interfaces.
- If transport profile is enabled on a tunnel, crypto maps are not supported on the tunnel source interfaces.
- Crypto maps are not supported on tunnel interface of MFR.
- Crypto maps are not supported on Vlan interfaces
- GetVPN crypto map is supported on port-channel interfaces.

Information About Configuring Security for VPNs with IPsec

Supported Standards

Cisco implements the following standards with this feature:

- IPsec—IPsec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer; IPsec uses IKE to handle negotiation of protocols and algorithms based on the local policy, and generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Note The term IPsec is sometimes used to describe the entire protocol of IPsec data services and IKE security protocols, and is also sometimes used to describe only the data services.

- IKE (IKEv1 and IKEv2)—A hybrid protocol that implements Oakley and SKEME key exchanges inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. While IKE is used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of IPsec peers, negotiates IPsec security associations, and establishes IPsec keys.

Note Starting from Cisco IOS XE Bengaluru 17.6.x, configuring a weak crypto algorithm generates a warning, but the warning can be safely ignored and does not impact the working of the algorithms. The following example displays a warning message for a weak crypto algorithm:

```
Device(config-ikev2-proposal)# group 5
%Warning: weaker dh-group is deprecated
```

The following table lists all the weak algorithms.

IKEv1	IKEv2	IPsec
DH_GROUP_768_MODP/Group 1	DH_GROUP_768_MODP/Group 1	ah-md5-hmac
DH_GROUP_1024_MODP/Group 2	DH_GROUP_1024_MODP/Group 2	ah-sha-hmac
DH_GROUP_1536_MODP/Group 5	DH_GROUP_1536_MODP/Group 5	esp-des
DES	DES	esp-3des
3DES	3DES	esp-sha-hmac
MD5	MD5	esp-gmac
DH_GROUP_2048_256_MODP/Group 24	DH_GROUP_2048_256_MODP/Group 24	esp-md5-hmac

IKEv1	IKEv2	IPsec
		esp-null

The component technologies implemented for IPsec include:

Note Starting from Cisco IOS XE 17.11.1a, as part of security hardening and deprecation of weak ciphers, the options to configure DES, 3DES, MD5, and Diffie-Hellman (DH) groups 1, 2, and 5 are deprecated and are no longer supported. Instead, use AES, SHA, and DH Groups 14 or higher. Additionally, the esp-gmac transforms are also deprecated.

If you want to continue using the weak ciphers, disable CSDL compliance on the device using the **crypto engine compliance shield disable** command, and reboot.

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is a privacy transform for IPsec and IKE and has been developed to replace DES. AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco software implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet. For backwards compatibility, Cisco IOS IPsec also implements the RFC 1829 version of ESP DES-CBC.

Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Cisco no longer recommends Triple DES (3DES).

Note Cisco IOS images with strong encryption (including, but not limited to 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send an e-mail to export@cisco.com.

- SHA-2 and SHA-1 family (HMAC variant)—Secure Hash Algorithm (SHA) 1 and 2. Both SHA-1 and SHA-2 are hash algorithms used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. SHA-2 family adds the SHA-256 bit hash algorithm and SHA-384 bit hash algorithm. This functionality is part of the Suite-B requirements that comprises four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.
- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. It supports 768-bit (the default), 1024-bit, 1536-bit, 2048-bit, 3072-bit, and 4096-bit DH groups. It also supports a 2048-bit DH group with a 256-bit subgroup, and 256-bit and 384-bit elliptic curve DH (ECDH). Cisco recommends using 2048-bit or larger DH key exchange, or ECDH key exchange.

- MD5 (Hash-based Message Authentication Code (HMAC) variant)—Message digest algorithm 5 (MD5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPsec as implemented in Cisco software supports the following additional standards:

- AH—Authentication Header. A security protocol, which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).
- ESP—Encapsulating Security Payload. A security protocol, which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

Supported Encapsulation

IPsec works with the following serial encapsulations: Frame Relay, High-Level Data-Links Control (HDLC), and PPP.

IPsec also works with Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Data Link Switching+ (DLSw+), and Source Route Bridging (SRB) tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPsec.

IPsec Functionality Overview

IPsec provides the following network security services. (In general, the local security policy dictates the use of one or more of these services.)

- Data confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPsec receiver can authenticate the source of the sent IPsec packets. This service is dependent upon the data integrity service.
- Anti-replay—The IPsec receiver can detect and reject replayed packets.

IPsec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer recognizes a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (The use of the term *tunnel* in this chapter does not refer to using IPsec in tunnel mode.)

More accurately, these *tunnels* are sets of security associations (SAs) that are established between two IPsec peers. The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams only need to be authenticated, while other data streams must both be encrypted and authenticated.

IKEv1 Transform Sets

An Internet Key Exchange version 1 (IKEv1) transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

IKEv2 Transform Sets

An Internet Key Exchange version 2 (IKEv2) proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in the negotiation. The default proposal is a collection of commonly used algorithms which are as follows:

```
encryption aes-cbc-128 3des
integrity sha1 md5
group 5 2
```

Although the **crypto ikev2 proposal** command is similar to the **crypto isakmp policy priority** command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuration of one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.

Note To use IKEv2 proposals in negotiation, they must be attached to IKEv2 policies. If a proposal is not configured, then the default IKEv2 proposal is used with the default IKEv2 policy.

Transform Sets: A Combination of Security Protocols and Algorithms

About Transform Sets

Note Cisco no longer recommends using ah-md5-hmac, esp-md5-hmac, esp-des or esp-3des. Instead, you should use ah-sha-hmac, esp-sha-hmac or esp-aes. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

A transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

During IPsec security association negotiations with IKE, peers search for an identical transform set for both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers' IPsec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

The table below shows allowed transform combinations.

Table 209: Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform (*Pick only one.*)	**ah-md5-hmac**	AH with the MD5 (Message Digest 5) (an HMAC variant) authentication algorithm. (No longer recommended).
	ah-sha-hmac	AH with the SHA (Secure Hash Algorithm) (an HMAC variant) authentication algorithm.
ESP Encryption Transform (*Pick only one.*)	**esp-aes**	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.
	esp-aes 192	ESP with the 192-bit AES encryption algorithm.
	esp-aes 256	ESP with the 256-bit AES encryption algorithm.
	esp-des	ESP with the 56-bit Data Encryption Standard (DES) encryption algorithm. (No longer recommended).
esp-3des		ESP with the 168-bit DES encryption algorithm (3DES or Triple DES). (No longer recommended).
ESP Authentication Transform (*Pick only one.*)	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm. (No longer recommended).
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Cisco IOS Suite-B Support for IKE and IPsec Cryptographic Algorithms

Suite-B has the following cryptographic algorithms:

- Suite-B-GCM-128-Provides ESP integrity protection, confidentiality, and IPsec encryption algorithms that use the 128-bit AES using Galois and Counter Mode (AES-GCM) described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.

- Suite-B-GCM-256-Provides ESP integrity protection and confidentiality using 256-bit AES-GCM described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.
- Suite-B-GMAC-128-Provides ESP integrity protection using 128-bit AES- Galois Message Authentication Code (GMAC) described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.
- Suite-B-GMAC-256-Provides ESP integrity protection using 256-bit AES-GMAC described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.

IPSec encryption algorithms use AES-GCM when encryption is required and AES-GMAC for message integrity without encryption.

IKE negotiation uses AES Cipher Block Chaining (CBC) mode to provide encryption and Secure Hash Algorithm (SHA)-2 family containing the SHA-256 and SHA-384 hash algorithms, as defined in RFC 4634, to provide the hash functionality. Diffie-Hellman using Elliptic Curves (ECP), as defined in RFC 4753, is used for key exchange and the Elliptic Curve Digital Signature Algorithm (ECDSA), as defined in RFC 4754, to provide authentication.

Suite-B Requirements

Suite-B imposes the following software crypto engine requirements for IKE and IPsec:

- HMAC-SHA256 and HMAC-SHA384 are used as pseudorandom functions; the integrity check within the IKE protocol is used. Optionally, HMAC-SHA512 can be used.
- Elliptic curve groups 19 (256-bit ECP curve) and 20 (384-bit ECP curve) are used as the Diffie-Hellman group in IKE. Optionally, group 21 (521-bit ECP curve) can be used.
- The Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- GCM (16 byte ICV) and GMAC is used for ESP (128-bit and 256-bit keys). Optionally, 192-bit keys can be used.
- Public Key Infrastructure (PKI) support for validation of X.509 certificates using ECDSA signatures must be used.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS must be used.
- IKEV2 support for allowing the ECDSA signature (ECDSA-sig) as authentication method must be used.

Where to Find Suite-B Configuration Information

Suite-B configuration support is described in the following documents:

- For more information on SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration, see the *Configuring Internet Key Exchange for IPsec VPNs* feature module.
- For more information on configuring a transform for an integrity algorithm type, see the “Configuring the IKEv2 Proposal” section in the *Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site* feature module.

- For more information on configuring the ECDSA-sig to be the authentication method for IKEv2, see the "Configuring IKEv2 Profile (Basic)" section in the *Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site* feature module.
- For more information on configuring elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation, see the *Configuring Internet Key Exchange for IPsec VPNs* and *Configuring Internet Key Exchange Version 2 and FlexVPN* feature modules.

For more information on the Suite-B support for certificate enrollment for a PKI, see the *Configuring Certificate Enrollment for a PKI* feature module.

How to Configure IPsec VPNs

Creating Crypto Access Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]
   - **ip access-list extended** *name*
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] • **ip access-list extended** *name* **Example:** `Device(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255`	Specifies conditions to determine which IP packets are protected. • You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list. • Enable or disable crypto for traffic that matches these conditions.

	Command or Action	Purpose
	Example: `Device(config)# ip access-list extended vpn-tunnel`	**Tip** Cisco recommends that you configure "mirror image" crypto access lists for use by IPsec and that you avoid using the **any** keyword.
Step 4	Repeat Step 3 for each crypto access list you want to create.	—

What to Do Next

After at least one crypto access list is created, a transform set needs to be defined as described in the "Configuring Transform Sets for IKEv1 and IKEv2 Proposals, on page 1914" section.

Next the crypto access lists need to be associated to particular interfaces when you configure and apply crypto map sets to the interfaces. (Follow the instructions in the "Creating Crypto Map Sets, on page 1918" and "Applying Crypto Map Sets to Interfaces, on page 1927" sections).

Configuring Transform Sets for IKEv1 and IKEv2 Proposals

Perform this task to define a transform set that is to be used by the IPsec peers during IPsec security association negotiations with IKEv1 and IKEv2 proposals.

Restrictions

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have a hardware IPsec encryption.
- Your router and the other peer must support IPsec.
- Your router and the other peer must support the k9 subsystem.
- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.
- Unlike IKEv1, the authentication method and SA lifetime are not negotiable in IKEv2, and because of this, these parameters cannot be configured under the IKEv2 proposal.

Configuring Transform Sets for IKEv1

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]
4. **mode** [**tunnel** | **transport**]
5. **end**
6. **clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*]
7. **show crypto ipsec transform-set** [**tag** *transform-set-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]] **Example:** `Device(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac`	Defines a transform set and enters crypto transform configuration mode. • There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and the table in "About Transform Sets" section provides a list of allowed transform combinations.
Step 4	**mode** [**tunnel** | **transport**] **Example:** `Device(cfg-crypto-tran)# mode transport`	(Optional) Changes the mode associated with the transform set. • The mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 5	**end** **Example:** `Device(cfg-crypto-tran)# end`	Exits crypto transform configuration mode and enters privileged EXEC mode.
Step 6	**clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*] **Example:** `Device# clear crypto sa`	(Optional) Clears existing IPsec security associations so that any changes to a transform set takes effect on subsequently established security associations. Manually established SAs are reestablished immediately. • Using the **clear crypto sa** command without parameters clears out the full SA database, which clears out active security sessions. • You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.
Step 7	**show crypto ipsec transform-set** [**tag** *transform-set-name*] **Example:** `Device# show crypto ipsec transform-set`	(Optional) Displays the configured transform sets.

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the*Creating Crypto Map Sets* section.

Configuring Transform Sets for IKEv2

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 proposal** *proposal-name*
4. **encryption** *transform1* [*transform2*] ...
5. **integrity** *transform1* [*transform2*] ...
6. **group** *transform1* [*transform2*] ...
7. **end**
8. **show crypto ikev2 proposal**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 proposal** *proposal-name* **Example:** `Device(config)# crypto ikev2 proposal proposal-1`	Specifies the name of the proposal and enters crypto IKEv2 proposal configuration mode. • The proposals are referred in IKEv2 policies through the proposal name.
Step 4	**encryption** *transform1* [*transform2*] ... **Example:** `Device(config-ikev2-proposal)# encryption aes-cbc-128`	(Optional) Specifies one or more transforms of the following encryption type: • AES-CBC 128—128-bit AES-CBC • AES-CBC 192—192-bit AES-CBC • AES-CBC 256—256-bit AES-CBC • 3DES—168-bit DES (No longer recommended. AES is the recommended encryption algorithm).
Step 5	**integrity** *transform1* [*transform2*] ... **Example:**	(Optional) Specifies one or more transforms of the following integrity type:

	Command or Action	Purpose
	`Device(config-ikev2-proposal)# integrity sha1`	• The **sha256** keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm. • The **sha384** keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm. • The **sha512** keyword specifies SHA-2 family 512-bit (HMAC variant) as the hash algorithm • the **sha1** keyword specifies the SHA-1 (HMAC variant) as the hash algorithm. • The **md5** keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended. SHA-1 is the recommended replacement.)
Step 6	**group** *transform1* [*transform2*] ... **Example:** `Device(config-ikev2-proposal)# group 14`	(Optional) Specifies one or more transforms of the possible DH group type: • **1**—768-bit DH (No longer recommended.) • **2**—1024-bit DH (No longer recommended) • **5**—1536-bit DH (No longer recommended) • **14**—Specifies the 2048-bit DH group. • **15**—Specifies the 3072-bit DH group. • **16**—Specifies the 4096-bit DH group. • **19**—Specifies the 256-bit elliptic curve DH (ECDH) group. • **20**—Specifies the 384-bit ECDH group. • **24**—Specifies the 2048-bit DH/DSA group.
Step 7	**end** **Example:** `Device(config-ikev2-proposal)# end`	Exits crypto IKEv2 proposal configuration mode and returns to privileged EXEC mode.
Step 8	**show crypto ikev2 proposal** **Example:** `Device# show crypto ikev2 proposal`	(Optional) Displays the parameters for each IKEv2 proposal.

Transform Sets for IKEv2 Examples

The following examples show how to configure a proposal:

IKEv2 Proposal with One Transform for Each Transform Type

```
Device(config)# crypto ikev2 proposal proposal-1
```

```
Device(config-ikev2-proposal)# encryption aes-cbc-128
Device(config-ikev2-proposal)# integrity sha1
Device(config-ikev2-proposal)# group 14
```

IKEv2 Proposal with Multiple Transforms for Each Transform Type

```
crypto ikev2 proposal proposal-2
encryption aes-cbc-128 aes-cbc-192
integrity sha1 sha256
group 14 15
```

For a list of transform combinations, see Configuring Security for VPNs with IPsec.

IKEv2 Proposals on the Initiator and Responder

The proposal of the initiator is as follows:

```
Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-196
Device(config-ikev2-proposal)# integrity sha1 sha256
Device(config-ikev2-proposal)# group 14 16
```

The proposal of the responder is as follows:

```
Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-196 aes-cbc-128
Device(config-ikev2-proposal)# integrity sha256 sha1
Device(config-ikev2-proposal)# group 16 14
```

In the scenario, the initiator's choice of algorithms is preferred and the selected algorithms are as follows:

```
encryption aes-cbc-128
integrity sha1
group 14
```

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the *Creating Crypto Map Sets* section.

Creating Crypto Map Sets

Creating Static Crypto Maps

When IKE is used to establish SAs, the IPsec peers can negotiate the settings they use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Perform this task to create crypto map entries that use IKE to establish SAs. To create IPv6 crypto map entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp**]
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*}
7. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
8. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**}
9. **set security-association level per-host**
10. **set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**]
11. **end**
12. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp**] **Example:** `Device(config)# crypto map static-map 1 ipsec-isakmp`	Creates or modifies a crypto map entry, and enters crypto map configuration mode. • For IPv4 crypto maps, use the command without the **ipv6** keyword.
Step 4	**match address** *access-list-id* **Example:** `Device(config-crypto-m)# match address vpn-tunnel`	Names an extended access list. • This access list determines the traffic that should be protected by IPsec and the traffic that should not be protected by IPsec security in the context of this crypto map entry.
Step 5	**set peer** {*hostname*	*ip-address*} **Example:**	Specifies a remote IPsec peer—the peer to which IPsec protected traffic can be forwarded.

	Command or Action	Purpose
	`Device(config-crypto-m)# set-peer 192.168.101.1`	• Repeat for multiple remote peers.
Step 6	**crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:** `Device(config-crypto-m)# set security-association dummy seconds 5`	Enables generating dummy packets. These dummy packets are generated for all flows created in the crypto map.
Step 7	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Device(config-crypto-m)# set transform-set aesset`	Specifies the transform sets that are allowed for this crypto map entry. • List multiple transform sets in the order of priority (highest priority first).
Step 8	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**} **Example:** `Device (config-crypto-m)# set security-association lifetime seconds 2700`	(Optional) Specifies a SA lifetime for the crypto map entry. • By default, the SAs of the crypto map are negotiated according to the global lifetimes, which can be disabled.
Step 9	**set security-association level per-host** **Example:** `Device(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate SAs should be established for each source and destination host pair. • By default, a single IPsec "tunnel" can carry traffic for multiple source hosts and multiple destination hosts. **Caution** Use this command with care because multiple streams between given subnets can rapidly consume resources.
Step 10	**set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**] **Example:** `Device(config-crypto-m)# set pfs group14`	(Optional) Specifies that IPsec either should ask for password forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPsec peer. • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier.

	Command or Action	Purpose
		• Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, group 1 is used as the default.
Step 11	**end** **Example:** `Device(config-crypto-m)# end`	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 12	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Device# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears out the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the Applying Crypto Map Sets to Interfaces, on page 1927 section.

Creating Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPsec SAs can be established. A dynamic crypto map entry that does not specify an access list is ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify the acceptable transform sets.

Perform this task to create dynamic crypto map entries that use IKE to establish the SAs.

Note IPv6 addresses are not supported on dynamic crypto maps.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
4. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
5. **match address** *access-list-id*
6. **set peer** {*hostname* | *ip-address*}
7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**}
8. **set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**]
9. **exit**
10. **exit**
11. **show crypto dynamic-map** [**tag** *map-name*]
12. **configure terminal**
13. **crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**]
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Device(config)# crypto dynamic-map test-map 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Device(config-crypto-m)# set transform-set aesset`	Specifies the transform sets allowed for the crypto map entry. • List multiple transform sets in the order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.

	Command or Action	Purpose
Step 5	**match address** *access-list-id* **Example:** `Device(config-crypto-m)# match address 101`	(Optional) Specifies the list number or name of an extended access list. • This access list determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec security in the context of this crypto map entry. **Note** Although access lists are optional for dynamic crypto maps, they are highly recommended. • If an access list is configured, the data flow identity proposed by the IPsec peer must fall within a **permit** statement for this crypto access list. • If an access list is not configured, the device accepts any data flow identity proposed by the IPsec peer. However, if an access list is configured but the specified access list does not exist or is empty, the device drops all packets. This is similar to static crypto maps, which require access lists to be specified. • Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation. • You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)
Step 6	**set peer** {*hostname* | *ip-address*} **Example:** `Device(config-crypto-m)# set peer 192.168.101.1`	(Optional) Specifies a remote IPsec peer. Repeat this step for multiple remote peers. **Note** This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**} **Example:** `Device(config-crypto-m)# set security-association lifetime seconds 7200`	(Optional) Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security SAs. **Note** To minimize the possibility of packet loss when rekeying in high bandwidth environments, you can disable the rekey request triggered by a volume lifetime expiry.
Step 8	**set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**]	(Optional) Specifies that IPsec should ask for PFS when requesting new security associations for this crypto map

	Command or Action	Purpose
	Example: `Device(config-crypto-m)# set pfs group14`	entry or should demand PFS in requests received from the IPsec peer. • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default.
Step 9	**exit** **Example:** `Device(config-crypto-m)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 11	**show crypto dynamic-map** [**tag** *map-name*] **Example:** `Device# show crypto dynamic-map`	(Optional) Displays information about dynamic crypto maps.
Step 12	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 13	**crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**] **Example:** `Device(config)# crypto map static-map 1 ipsec-isakmp dynamic test-map discover`	(Optional) Adds a dynamic crypto map to a crypto map set. • You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set.

	Command or Action	Purpose
		Note You must enter the **discover** keyword to enable TED.
Step 14	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the entire SA database must be reserved for large-scale changes, or when the router is processing minimal IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the "Applying Crypto Map Sets to Interfaces, on page 1927" section.

Creating Crypto Map Entries to Establish Manual SAs

Perform this task to create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs). To create IPv6 crypto maps entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** [**ipv6**] *map-name seq-num* [**ipsec-manual**]
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name*
7. Do one of the following:
   - **set session-key inbound ah** *spi hex-key-string*
   - **set session-key outbound ah** *spi hex-key-string*
8. Do one of the following:
   - **set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
   - **set session-key outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
9. **exit**
10. **exit**

11. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-manual**] **Example:** `Device(config)# crypto map mymap 10 ipsec-manual`	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode. • For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.
Step 4	**match address** *access-list-id* **Example:** `Device(config-crypto-m)# match address 102`	Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry. • The access list can specify only one **permit** entry when IKE is not used.
Step 5	**set peer** {*hostname* | *ip-address*} **Example:** `Device(config-crypto-m)# set peer 10.0.0.5`	Specifies the remote IPsec peer. This is the peer to which IPsec protected traffic should be forwarded. • Only one peer can be specified when IKE is not used.
Step 6	**set transform-set** *transform-set-name* **Example:** `Device(config-crypto-m)# set transform-set someset`	Specifies which transform set should be used. • This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. **Note** Only one transform set can be specified when IKE is not used.

Step 7	Do one of the following:
	• **set session-key inbound ah** *spi hex-key-string*
	• **set session-key outbound ah** *spi hex-key-string*
	Example:
	`Device(config-crypto-m)# set session-key inbound ah 256 98765432109876549876543210987654`
	Example:
	`Device(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc`
		Sets the AH security parameter indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol.
		• This manually specifies the AH security association to be used with protected traffic.

	Command or Action	Purpose

Step 8	Do one of the following:
	• **set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
	• **set session-key outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
	Example:
	`Device(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345`
	Example:
	`Device(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd`
		Sets the Encapsulating Security Payload (ESP) Security Parameter Indexes (SPI) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol.
		Or
		Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm.
		• This manually specifies the ESP security association to be used with protected traffic.

Step 9	**exit** **Example:** `Device(config-crypto-m)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 11	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Device# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the entire SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the "Applying Crypto Map Sets to Interfaces, on page 1927" section.

Applying Crypto Map Sets to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic flows. Applying the crypto map set to an interface instructs the device to evaluate the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by the crypto map.

Perform this task to apply a crypto map to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number*
4. **crypto map** *map-name*
5. **exit**
6. **crypto map** *map-name* **local-address** *interface-id*
7. **exit**
8. **show crypto map** [**interface** *interface*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/number* **Example:** `Device(config)# interface FastEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 4	**crypto map** *map-name* **Example:** `Device(config-if)# crypto map mymap`	Applies a crypto map set to an interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**crypto map** *map-name* **local-address** *interface-id* **Example:** `Device(config)# crypto map mymap local-address loopback0`	(Optional) Permits redundant interfaces to share the same crypto map using the same local identity.
Step 7	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode.
Step 8	**show crypto map** [**interface** *interface*] **Example:** `Device# show crypto map`	(Optional) Displays your crypto map configuration

Configuration Examples for IPsec VPN

Example: Configuring AES-Based Static Crypto Map

This example shows how a static crypto map is configured and how an AES is defined as the encryption method:

```
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 group 14
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
!
!
voice call carrier capacity active
!
!
mta receive maximum-recipients 0
!
!
interface FastEthernet0/0
 ip address 10.0.110.2 255.255.255.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map aesmap
!
interface Serial0/0
 no ip address
 shutdown
!
interface FastEthernet0/1
 ip address 10.0.110.1 255.255.255.0
 ip nat inside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip nat inside source list 110 interface FastEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.5.1.1
ip route 10.0.110.0 255.255.255.0 FastEthernet0/0
ip route 172.18.124.0 255.255.255.0 10.5.1.1
ip route 172.18.125.3 255.255.255.255 10.5.1.1
ip http server
```

```
!
!
access-list 110 deny   ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
access-list 110 permit ip 10.0.110.0 0.0.0.255 any
access-list 120 permit ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
!
```

Additional References for Configuring Security for VPNs with IPsec

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IKE, IPsec, and PKI configuration commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IKE configuration	*Configuring Internet Key Exchange for IPsec VPNs*
Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration	*Configuring Internet Key Exchange for IPsec VPNs*
Suite-B Integrity algorithm type transform configuration	*Configuring Internet Key Exchange Version 2 (IKEv2)*
Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2	*Configuring Internet Key Exchange Version 2 (IKEv2)*
Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	• *Configuring Internet Key Exchange for IPsec VPNs* • *Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site*
Suite-B support for certificate enrollment for a PKI	*Configuring Certificate Enrollment for a PKI*
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR-MIB • CISCO-IPSEC-MIB • CISCO-IPSEC-POLICY-MAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2402	*IP Authentication Header*
RFC 2403	*The Use of HMAC-MD5-96 within ESP and AH*
RFC 2404	*The Use of HMAC-SHA-1-96 within ESP and AH*
RFC 2405	*The ESP DES-CBC Cipher Algorithm With Explicit IV*
RFC 2406	*IP Encapsulating Security Payload (ESP)*
RFC 2407	*The Internet IP Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol (ISAKMP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Security for VPNs with IPsec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 210: Feature Information for Configuring Security for IPsec VPNs

Feature Name	Software Releases	Feature Information
Advanced Encryption Standard		This feature adds support for the new encryption standard AES, which is a privacy transform for IPsec and IKE and has been developed to replace DES. The following commands were modified by this feature: **crypto ipsec transform-set**, **encryption (IKE policy)**, **show crypto ipsec transform-set**, **show crypto isakmp policy**.
Suite-B Support in IOS SW Crypto		Suite-B adds support for four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. The following command was modified by this feature: **crypto ipsec transform-set**.

Note GetVPN crypto map is supported on port-channel interfaces from IOS XE 16.9.1 onwards.

Glossary

anti-replay—Security service where the receiver can reject old or duplicate packets to protect itself against replay attacks. IPsec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS XE IPsec provides this service whenever it provides the data authentication service, except for manually established SAs (that is, SAs established by configuration and not by IKE).

data authentication—Verification of the integrity and origin of the data. Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

data confidentiality—Security service in which the protected data cannot be observed.

data flow—Grouping of traffic, identified by a combination of source address or mask, destination address or mask, IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. IPsec protection is applied to data flows.

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPsec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

peer—In the context of this module, a "peer" is a router or other device that participates in IPsec.

PFS—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Description of how two or more entities use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

SPI—security parameter index. A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. Without IKE, the SPI is manually specified for each security association.

transform—List of operations performed on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

tunnel—In the context of this module, "tunnel" is a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.

CHAPTER 157

IPsec Virtual Tunnel Interfaces

IPsec virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify the configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Restrictions for IPsec Virtual Tunnel Interfaces

Fragmentation

Fragmentation is not supported over IPsec tunnel. You can choose to set the lower MTU on hosts to avoid packet fragments or choose to fragment the packets on any device.

IPsec Transform Set

The IPsec transform set must be configured in tunnel mode only.

IKE Security Association

The Internet Key Exchange (IKE) security association (SA) is bound to the VTI.

IPsec SA Traffic Selectors

Static VTIs (SVTIs) support only a single IPsec SA that is attached to the VTI interface. The traffic selector for the IPsec SA is always “IP any any.”

By default, Static VTIs (SVTIs) support only a single IPSec SA that is attached to the virtual tunnel interface. The traffic selector for the IPsec SA is always “IP any any”.

IPv4

This feature supports SVTIs that are configured to encapsulate IPv4 packets .

Tunnel Protection

Do not configure the **shared** keyword when using the **tunnel mode ipsec ipv4** command for IPsec IPv4 mode.

Traceroute

The traceroute function with crypto offload on VTIs is not supported.

VxLAN GPE Tunnel Interface

The VxLAN GPE Tunnel Interface cannot use the same source interface as IPsec VTI.

Information About IPsec Virtual Tunnel Interfaces

The use of IPsec VTIs can simplify the configuration process when you need to provide protection for remote access and it provides an alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation. A benefit of using IPsec VTIs is that the configuration does not require static mapping of IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel.

The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration . Because DVTIs function like any other real interface you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

The following sections provide details about the IPSec VTI:

Benefits of Using IPsec Virtual Tunnel Interfaces

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface. When IPsec VTIs are used, you can separate the application of features such as Network Address Translation (NAT), ACLs, and QoS and apply them to clear-text, or encrypted text, or both.

There are two types of VTI interfaces: static VTIs (SVTIs) and dynamic VTIs (DVTIs).

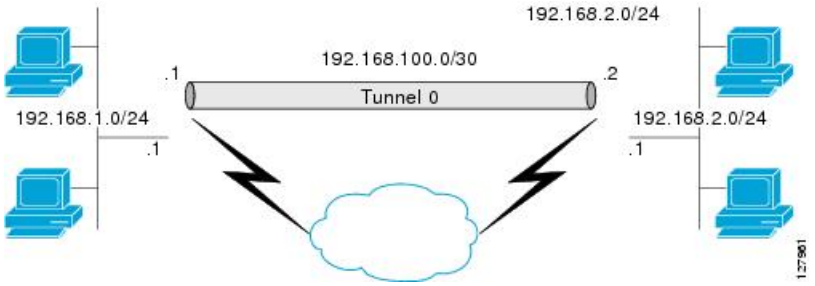
Static Virtual Tunnel Interfaces

SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites.

Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

The figure below illustrates how a SVTI is used.

Figure 76: IPsec SVTI



The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.

Multi-SA Support for SVTI

By default, the traffic selector for an SVTI is set to ‘any any’. As a result, a single IPSec SA is attached for the SVTI corresponding to the ‘any any’ traffic selector.

From Cisco IOS XE Gibraltar 16.12.1, you can define and associate an Access Control List (ACL) with an SVTI to select traffic between specific source and destination proxies instead of the ‘any any’ proxy defined by the default. IPSec SAs are created for each non-any-any traffic selector, and thus, multiple SAs are attached to an SVTI.

This feature is supports IPv4 and IPv6 traffic protection with IPSec encapsulation in tunnel mode. The feature supports both IKEv1 and IKEv2.

Restrictions

- This feature is not supported with tunnel protection shared.
- This feature is not supported with IPSec Mixed Mode.
- Traffic selectors associated with the SVTIs at both the ends of a tunnel must have matching source and destination proxies. Do not narrow down the traffic selector at one of the SVTIs forming a tunnel.

ACL Characteristics and Effects on SVTI IPSec SAs

- An ACL associated with an SVTI must not contain an ‘any any’ proxy. For an ‘any any’ traffic selector, use the default behaviour of the SVTI and do not associate an ACL with the SVTI.
- An ACL associated with an SVTI supports only **permit** statements and must not contain **deny** statements.
- Run-time modification of an ACL associated with an SVTI is not supported. Shut the tunnel down before adding or modifying ACEs in the ACL.
- If you disassociate an ACL from an SVTI, existing IPSec SAs are deleted and a new IPSec SA for default traffic selector of ‘IP any any’ is formed.

- We recommend that you associate a maximum of 100 Access Control Entries (ACEs) with an SVTI. Further, all the ACLs associated with the various tunnel interfaces should together use a maximum of 2000 ACEs.

Reverse Route Injection

For Multi-SA SVTIs, Reverse Route Injection (RRI) can be configured in the IPSec profile.

If you use extended ACL or ACE options, such as protocol, port number, and DHCP, do not use RRI; use other means such as route maps for routing.

Note RRI capability with distance and tag is yet to be supported.

Dual Stack Support for SVTI

SVTI Dual Stack feature provides the capabilities to carry both IPv4 and IPv6 traffic using a single IPsec Security Association (SA) that is tunnelled over IPv4. From IOS XE release 17.9 onwards, Cisco supports specific subnets in ACL when the ingress end of the tunnel interface is configured with a third party IPSec client. Also, based on the third party IPsec client configuration, it responds with a specific traffic selector. In this case, the IPsec supports non-any non-any proxy configuration and allows to carry IPv4 or IPv6 type of traffic in the tunnel interface. This feature is supported only with IKEv2.

Restrictions

- Tunnel-mode configuration is allowed only under the IPsec profiles when you use the tunnel interface in dual-overlay mode.
- In Cisco IOS XE, ACL filtering infrastructure does not work on traffic generated locally on the device.
- You have to use the same set of traffic selectors for rekeying an IPsec SA. You cannot change the traffic selectors during the rekey process but when you change, the rekey request is rejected with the message *TS_UNACCEPTABLE*.
- A maximum of 16 traffic selectors are accepted at the IKEv2 level.
- ACLs on dual-stack tunnel interface are not supported. Any ACL configured on this interface is overwritten by dual-stack ACLs.

Dynamic Virtual Tunnel Interfaces

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

Note You can configure DVTIs with IKEv1 or IKEv2. The legacy crypto map based configuration supports DVTIs with IKEv1 only. A DVTI configuration with IKEv2 is supported only in FlexVPN.

DVTIs can be used for both the server and the remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs.

DVTIs function like any other real interface, so you can apply QoS, firewall, or other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

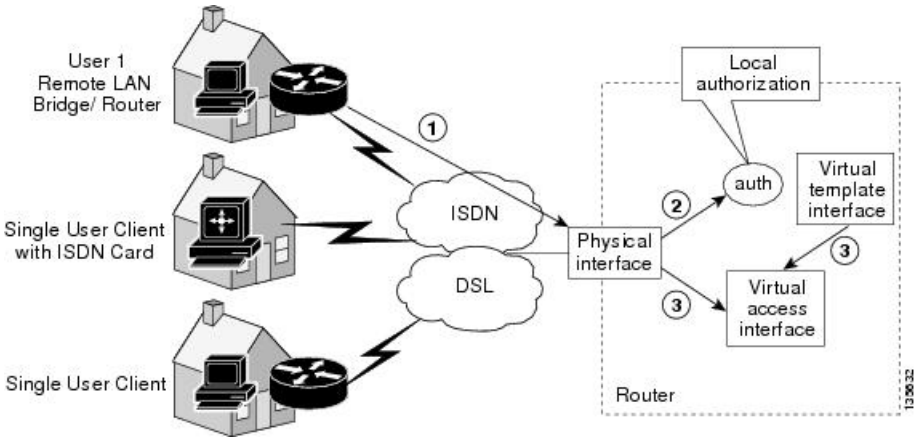
DVTIs provide efficiency in the use of IP addresses and provide secure connectivity. DVTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using an extended authentication (Xauth) User or Unity group, or can be derived from a certificate. DVTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec DVTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The DVTI simplifies VPN routing and forwarding- (VRF-) aware IPsec deployment. The VRF is configured on the interface.

A DVTI requires minimal configuration on the router. A single virtual template can be configured and cloned.

The DVTI creates an interface for IPsec sessions and uses the virtual template infrastructure for dynamic instantiation and management of dynamic IPsec VTIs. The virtual template infrastructure is extended to create dynamic virtual-access tunnel interfaces. DVTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs.

The figure below illustrates the DVTI authentication path.

Figure 77: Dynamic IPsec VTI



The authentication shown in the figure above follows this path:

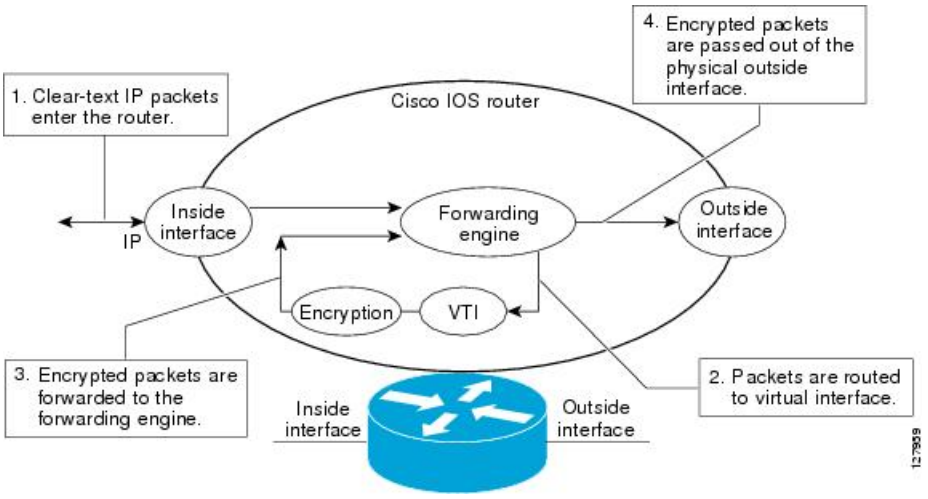
1. User 1 calls the router.
2. Router 1 authenticates User 1.
3. IPsec clones the virtual access interface from the virtual template interface.

Traffic Encryption with the IPsec Virtual Tunnel Interface

When an IPsec VTI is configured, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static routing can be used to route traffic to the SVTI. DVTI uses reverse route injection to further simplify the routing configurations. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration . The IPsec virtual tunnel also allows you to encrypt multicast traffic with IPsec.

IPsec packet flow into the IPSec tunnel is illustrated in the figure below.

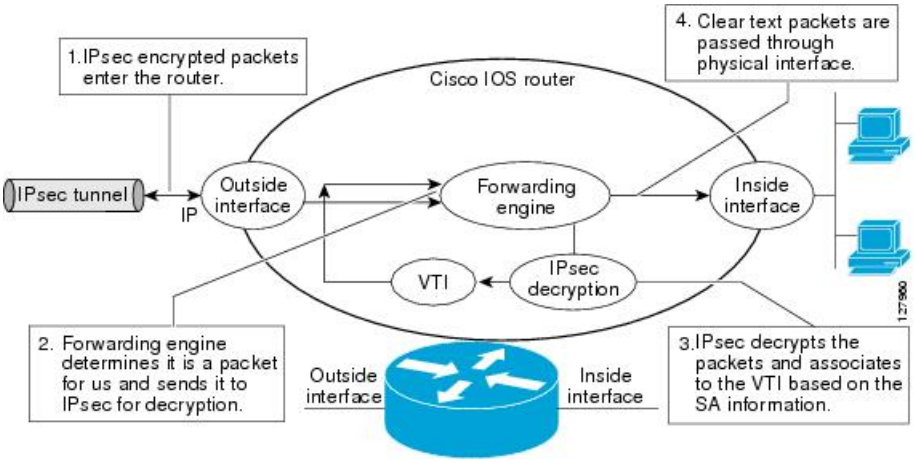
Figure 78: Packet Flow into the IPsec Tunnel



After packets arrive on the inside interface, the forwarding engine switches the packets to the VTI, where they are encrypted. The encrypted packets are handed back to the forwarding engine, where they are switched through the outside interface.

The figure below shows the packet flow out of the IPsec tunnel.

Figure 79: Packet Flow out of the IPsec Tunnel



Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define the policy for DVTIs. The dynamic interface is created at the end of IKE Phase 1 and IKE Phase 1.5. The interface is deleted when the IPsec session to the peer is closed. The IPsec session is closed when both IKE and IPsec SAs to the peer are deleted.

Routing with IPsec Virtual Tunnel Interfaces

Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as the tunnel endpoint. You can route to the interface or apply services such as QoS, firewalls, network address translation (NAT), and NetFlow statistics as you would to any other interface. You can monitor the interface and route to it, and the interface provides benefits similar to other Cisco IOS interface.

FlexVPN Mixed Mode Support

The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.

The FlexVPN Mixed Mode feature provides support for carrying IPv6 traffic over IPsec IPv4 transport from Cisco IOS XE Everest 16.4.1.

Auto Tunnel Mode Support in IPsec

When configuring a VPN headend in a multiple vendor scenario, you must be aware of the technical details of the peer or responder. For example, some devices may use IPsec tunnels while others may use generic routing encapsulation (GRE) or IPsec tunnel, and sometimes, a tunnel may be IPv4 or IPv6. In the last case, you must configure an Internet Key Exchange (IKE) profile and a virtual template.

The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. This feature is useful on dual stack hubs aggregating multivendor remote access, such as Cisco AnyConnect VPN Client, Microsoft Windows7 Client, and so on.

Note The Tunnel Mode Auto Selection feature eases the configuration for a responder only. The tunnel must be statically configured for an initiator.

IPSec Mixed Mode Support for VTI

The IPSec Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is supported for SVTI as well as DVTI and IKEv1 as well as IKEv2.

How to Configure IPsec Virtual Tunnel Interfaces

Configuring Static IPsec Virtual Tunnel Interfaces

Before you begin

Before configuring the tunnel protection for an IPsec profile, it is mandatory to shut down the tunnel interface. After configuration, enable the tunnel interface manually.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**
6. **interface** *type number*
7. **ip address** *address mask*
8. **tunnel mode ipsec ipv4**
9. **tunnel source** *interface-type interface-number*
10. **tunnel destination** *ip-address*
11. **tunnel protection IPsec profile** *profile-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto IPsec profile** *profile-name* **Example:** `Device(config)# crypto IPsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets can be used .

	Command or Action	Purpose
	Example: `Device(ipsec-profile)# set transform-set tset`	
Step 5	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile configuration mode, and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface tunnel 0`	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 7	**ip address** *address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Specifies the IP address and mask.
Step 8	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 9	**tunnel source** *interface-type interface-number* **Example:** `Device(config-if)# tunnel source loopback 0`	Specifies the tunnel source as a loopback interface.* **Note** *If you are configuring the Tunnel Mode Auto Selection feature using a virtual-template, omit the tunnel source and tunnel mode in interface virtual-template number type tunnel command. If the tunnel source and tunnel mode are specified, clients using IPv6 transport will fail to connect.
Step 10	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 172.16.1.1`	Identifies the IP address of the tunnel destination.
Step 11	**tunnel protection IPsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection IPsec profile PROF`	Associates a tunnel interface with an IPsec profile.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BGP over IPsec Virtual Tunnel Interfaces

Perform this task to optionally configure BGP over the virtual tunnel interfaces of two routers.

Before you begin

Perform steps in Configuring Static IPsec Virtual Tunnel Interfaces, on page 1942.

SUMMARY STEPS

1. **router bgp** *autonomous-system-number*
2. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
3. **network** *network-ip-address* **mask** *subnet-mask*
4. **exit**
5. Enter the following commands on the second router.
6. **router bgp** *autonomous-system-number*
7. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
8. **network** *network-ip-address* **mask** *subnet-mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65510`	Enters router configuration mode and creates a BGP routing process. *autonomous-system-number* —Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535. In the example, the first router in this procedure is identified as "65510".
Step 2	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.1.1.2 remote-as 65511`	*ip-address* —IP address of the adjacent router's tunnel interface. *autonomous-system-number* —Number of an autonomous system that identifies the router of the second router. Number in the range from 1 to 65535.
Step 3	**network** *network-ip-address* **mask** *subnet-mask* **Example:** `Device(config-router)# network 2.2.2.0 mask 255.255.255.0`	*network-ip-address*—IP address of the network advertised in BGP. For example, the IP address of a loopback interface. *subnet-mask*—subnet mask of the network advertised in BGP.

	Command or Action	Purpose
		Note The BGP network command network and mask *must* exactly match a route that is already in the routing table for it to be brought into BGP and advertised to BGP neighbors. This is different from EIGRP, OSPF where the network statement just has to "cover" an interface network and it will pick up the network with mask from the interface.
Step 4	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 5	Enter the following commands on the second router.	
Step 6	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65511`	Enters router configuration mode and creates a BGP routing process. *autonomous-system-number* —Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535. In the example, the second router in this procedure is identified as "65511".
Step 7	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.1.1.1 remote-as 65510`	*ip-address* —IP address of the adjacent router's tunnel interface.
Step 8	**network** *network-ip-address* **mask** *subnet-mask* **Example:** `Device(config-router)# network 1.1.1.0 mask 255.255.255.0`	*network-ip-address*—IP address of the network advertised in BGP. For example, the IP address of a loopback interface. *subnet-mask*—subnet mask of the network advertised in BGP. **Note** Use the exact network IP address and subnet mask.

Configuring Dynamic IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**

6. **interface virtual-template** *number* **type tunnel**
7. **tunnel mode ipsec ipv4**
8. **tunnel protection IPsec profile** *profile-name*
9. **exit**
10. **crypto isakamp profile** *profile-name*
11. **match identity address** *ip-address mask*
12. **virtual template** *template-number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *profile-name* **Example:** `Device(config)# crypto ipsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Device(ipsec-profile)# set transform-set tset`	Specifies which transform sets can be used with the crypto map entry.
Step 5	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits ipsec profile configuration mode and enters global configuration mode.
Step 6	**interface virtual-template** *number* **type tunnel** **Example:** `Device(config)# interface virtual-template 2 type tunnel`	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 7	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 8	**tunnel protection IPsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec profile PROF`	Associates a tunnel interface with an IPsec profile.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 10	**crypto isakamp profile** *profile-name* **Example:** `Device(config)# crypto isakamp profile profile1`	Defines the ISAKMP profile to be used for the virtual template.
Step 11	**match identity address** *ip-address mask* **Example:** `Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0`	Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.
Step 12	**virtual template** *template-number* **Example:** `Device(config)# virtual-template 1`	Specifies the virtual template attached to the ISAKMP profile.
Step 13	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring Multi-SA Support for Dynamic Virtual Tunnel Interfaces Using IKEv1

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **crypto keyring** *keyring-name*
7. **pre-shared-key** *address* **key** *key*
8. **exit**
9. **crypto isakmp profile** *profile-name*
10. **keyring** *keyring-name*
11. **match identity** *address mask*

12. **virtual-template** *template-number*
13. **exit**
14. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*]
15. **exit**
16. **crypto ipsec profile** *name*
17. **set security-policy limit** *maximum-limit*
18. **set transform-set** *transform-set-name* [*transform-set-name2 .... transform-set-name6*]
19. **exit**
20. **interface virtual-template** *number type tunnel*
21. **ip vrf forwarding** *vrf-name*
22. **ip unnumbered** *type number*
23. **tunnel mode ipsec ipv4**
24. **tunnel protection profile ipsec** *profile-name*
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf VRF-100-1`	Defines the VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:21`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 6	**crypto keyring** *keyring-name* **Example:** `Device(config)# crypto keyring cisco-100-1`	Defines a crypto key ring and enters key ring configuration mode.
Step 7	**pre-shared-key** *address* **key** *key* **Example:** `Device(config-keyring)# pre-shared-key address 10.1.1.1 key cisco-100-1`	Defines the preshared key to be used for Internet Key Exchange (IKE) authentication.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(config-keyring)# exit`	Exits keyring configuration mode and enters global configuration mode.
Step 9	**crypto isakmp profile** *profile-name* **Example:** `Device(config)# crypto isakmp profile cisco-isakmp-profile-100-1`	Defines an ISAKMP profile and enters ISAKMP configuration mode.
Step 10	**keyring** *keyring-name* **Example:** `Device(conf-isa-prof)# keyring cisco-100-1`	Configures a key ring in ISAKMP mode.
Step 11	**match identity** *address mask* **Example:** `Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0`	Matches an identity from the ISAKMP profile.
Step 12	**virtual-template** *template-number* **Example:** `Device(conf-isa-prof)# virtual-template 101`	Specifies the virtual template that will be used to clone virtual access interfaces.
Step 13	**exit** **Example:** `Device(conf-isa-prof)# exit`	Exits ISAKMP profile configuration mode and enters global configuration mode.
Step 14	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] **Example:** `Device(config)# crypto ipsec transform-set cisco esp-aes esp-sha-hmac`	Defines the transform set and enters crypto transform configuration mode.
Step 15	**exit** **Example:** `Device(conf-crypto-trans)# exit`	Exits crypto transform configuration mode and enters global configuration mode.
Step 16	**crypto ipsec profile** *name* **Example:** `Device(config)# crypto ipsec profile cisco-ipsec-profile-101`	Defines the IPsec parameters used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 17	**set security-policy limit** *maximum-limit* **Example:** `Device(ipsec-profile)# set security-policy limit 3`	Defines an upper limit to the number of flows that can be created for an individual virtual access interface.

	Command or Action	Purpose
Step 18	**set transform-set** *transform-set-name* [*transform-set-name2 .... transform-set-name6*] **Example:** `Device(ipsec-profile)# set transform-set cisco`	Specifies the transform sets to be used with the crypto map entry.
Step 19	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile and enters global configuration mode.
Step 20	**interface virtual-template** *number type tunnel* **Example:** `Device(config)# interface virtual-template 101 type tunnel`	Creates a virtual template interface that can be configured interface and enters interface configuration mode.
Step 21	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding VRF-100-1`	Associates a VRF instance with a virtual-template interface.
Step 22	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered GigabitEthernet 0.0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 23	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 24	**tunnel protection profile ipsec** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec profile PROF`	Associates a tunnel interface with an IPsec profile.
Step 25	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode, and returns to privileged EXEC mode.

Configuring IPsec Mixed Mode Support for SVTIs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]

5. **exit**
6. **interface** *type number*
7. **ip address** *address mask*
8. Do one of the following:
   - **tunnel mode ipsec ipv4 v6-overlay**
   - **tunnel mode ipsec ipv6 v4-overlay**
9. **tunnel source** *interface-type interface-type*
10. **tunnel destination** *ip-address*
11. **tunnel protection IPsec profile** *profile-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto IPsec profile** *profile-name* **Example:** `Device(config)# crypto IPsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Device(ipsec-profile)# set transform-set tset`	Specifies which transform sets can be used with the crypto map entry.
Step 5	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile configuration mode, and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface tunnel 0`	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 7	**ip address** *address mask* **Example:**	Specifies the IP address and mask.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.1.1.1 255.255.255.0`	
Step 8	Do one of the following: • **tunnel mode ipsec ipv4 v6-overlay** • **tunnel mode ipsec ipv6 v4-overlay** **Example:** `Device(config-if)# tunnel mode ipsec ipv4 v6-overlay`	Defines the mode for the tunnel.
Step 9	**tunnel source** *interface-type interface-type* **Example:** `Device(config-if)# tunnel source loopback 0`	Specifies the tunnel source as a loopback interface.
Step 10	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 172.16.1.1`	Identifies the IP address of the tunnel destination.
Step 11	**tunnel protection IPsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection IPsec profile PROF`	Associates a tunnel interface with an IPsec profile.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring IPsec Mixed Mode Support for Dynamic VTIs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set mixed mode**
5. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
6. **exit**
7. **interface virtual-template** *number* **type tunnel**
8. **tunnel mode ipsec ipv4**
9. **tunnel protection IPsec profile** *profile-name*
10. **exit**
11. **crypto isakamp profile** *profile-name*

12. **match identity address** *ip-address mask*
13. **virtual template** *template-number*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *profile-name* **Example:** `Device(config)# crypto ipsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 4	**set mixed mode** **Example:** `Device(config)# set mixed mode`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 5	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Device(ipsec-profile)# set transform-set tset`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits ipsec profile configuration mode and enters global configuration mode.
Step 7	**interface virtual-template** *number* **type tunnel** **Example:** `Device(config)# interface virtual-template 2 type tunnel`	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 8	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 9	**tunnel protection IPsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec profile PROF`	Associates a tunnel interface with an IPsec profile.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 11	**crypto isakamp profile** *profile-name* **Example:** `Device(config)# crypto isakamp profile profile1`	Defines the ISAKMP profile to be used for the virtual template.
Step 12	**match identity address** *ip-address mask* **Example:** `Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0`	Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.
Step 13	**virtual template** *template-number* **Example:** `Device(config)# virtual-template 1`	Specifies the virtual template attached to the ISAKMP profile.
Step 14	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring Multi-SA Support for Static IPsec Virtual Tunnel Interfaces

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **crypto IPsec profile** *profile-name*

Example:

```
Device(config)# crypto IPsec profile PROF
```

Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.

Step 4 **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]

Example:

```
Device(ipsec-profile)# set transform-set tset
```

Specifies the transform sets that can be used.

Step 5 **exit**

Example:

```
Device(ipsec-profile)# exit
```

Exits IPsec profile configuration mode, and enters global configuration mode.

Step 6 **interface** *type number*

Example:

```
Device(config)# interface tunnel 0
```

Specifies the interface on which the tunnel will be configured and enters interface configuration mode.

Step 7 **ip address** *address mask*

Example:

```
Device(config-if)# ip address 10.1.1.1 255.255.255.0
```

Specifies the IP address and mask.

Step 8 **tunnel mode ipsec** {**ipv4** | **ipv6**}

Example:

```
Device(config-if)# tunnel mode ipsec ipv4
```

Defines the mode for the tunnel.

Step 9 **tunnel source** *interface-type interface-number*

Example:

```
Device(config-if)# tunnel source loopback 0
```

Specifies the tunnel source as a loopback interface.

Step 10 **tunnel destination** *ip-address*

Example:

```
Device(config-if)# tunnel destination 172.16.1.1
```

Identifies the IP address of the tunnel destination.

Step 11 **tunnel protection ipsec policy** {**ipv4** | **ipv6**} *acl*

Example:

```
Device(config-if)# tunnel protection ipsec policy ipv4 ipsec-acl1
```

Associates an ACL with an SVTI to define non-any-any traffic selectors.

Step 12 **tunnel protection ipsec profile** *profile-name*

Example:

```
Device(config-if)# tunnel protection IPsec profile PROF
```

Associates a tunnel interface with an IPsec profile.

Step 13 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and enters global configuration mode.

Step 14 **ip access-list extended** *name* OR **ipv6 access-list** *name*

Example:

IPv4:

```
Device(config)# ip access-list extended ipsec-acl1
```

IPv6:

```
Device(config)# ipv6 access-list ipsec-acl1
```

Defines an extended IP access list using a name and enters extended named access list configuration mode.

Step 15 **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*]

Example:

```
Device(config-ext-nacl)# permit ip 30.0.1.0 0.0.0.255 10.0.1.0 0.0.0.255
```

Permits traffic that matches all of the conditions specified in the statement.

Do not use the keyword **any** as the wildcard for both the source and destination proxies. For the 'any any' traffic selector, use the default SVTI without an attached ACL.

Do not use **deny** statements.

Step 16 **end**

Example:

```
Device(config-ext-nacl)# end
```

Exits standard named access list configuration mode and enters privileged EXEC mode.

Configuring Tunnel Mode as Dual-overlay

To configure the tunnel mode as dual-overlay, perform these steps:

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface tunnel** *type number*

Example:

```
Device(config)# interface tunnel 1
```

Specifies a tunnel interface and number, and enters interface configuration mode.

Step 4 **ipv6 enable**

Example:

```
Device(config-if)# ipv6 enable
```

Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

Step 5 **tunnel source** { **ipv4-address | interface-type | interface-number**}

Example:

```
Device(config-if)# tunnel source Gigabitethernet 1
```

Specifies the source IPv6 address or the source interface type and number for the tunnel interface. If an interface type and number are specified, that interface must be configured with an IPv6 address.

Step 6 **tunnel mode ipsec dual-overlay**

Example:

```
Device(config-if)# tunnel mode ipsec dual-overlay
```

Specifies a dual-overlay tunnel. The **tunnel mode ipsec dual-overlay** command specifies the encapsulation protocol for the tunnel.

Step 7 **tunnel destination ip address** *address mask*

Example:

```
Device(config-if)# tunnel destination 89.89.89.1 255.255.255.255.0
```

Specifies the destination IPv6 address for the tunnel interface.

Step 8 **tunnel protection ipsec profile** *ipserc profile-name*

Example:

```
Device(config-if)# tunnel protection IPsec profile ipsecprof
```

Associates a tunnel interface with an IPsec profile. The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto IPsec profile** *name* command

Step 9 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and enters global configuration mode.

Step 10 **end**

Example:

```
Device(config-if)# end
```

Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPsec Virtual Tunnel Interfaces

Example: Static Virtual Tunnel Interface with IPsec

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPsec VTI for encryption and then sent out the physical interface. The tunnel on subnet 10 checks packets for the IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. The figure below illustrates the IPsec VTI configuration.

Figure 80: VTI with IPsec

Router Configuration

```
version 12.3
service timestamps debug datetime
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
encr aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.203 255.255.255.0

 load-interval 30
 tunnel source 10.0.149.203
 tunnel destination 10.0.149.217
 tunnel mode IPsec ipv4
 tunnel protection IPsec profile P1
!

 ip address 10.0.149.203 255.255.255.0
 duplex full
!

 ip address 10.0.35.203 255.255.255.0
 duplex full
!
ip classless
ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

Router Configuration

```
version 12.3
hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0

 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
interface
 ip address 10.0.149.217 255.255.255.0
 speed 100
 full-duplex
!
interface
 ip address 10.0.36.217 255.255.255.0
 load-interval 30
 full-duplex
!
ip classless
ip route 10.0.35.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface

This section provides information that you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is “up,” and the line protocol is “up.” If the line protocol is “down,” the session is not active.

Verifying the IPsec Static Virtual Tunnel Interface

```
Router# show interface tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport ipsec/ip, key disabled, sequencing disabled
Tunnel TTL 255
```

```
Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPsec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out

Router# show crypto session

Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPsec FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4,
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

Example: VRF-Aware Static Virtual Tunnel Interface

To add the VRF to the static VTI example, include the **ipvrf** and **ip vrf forwarding** commands to the configuration as shown in the following example.

C8000 Router Configuration

```
hostname c8000
.
.
ip vrf sample-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
interface Tunnel0
```

```
 ip vrf forwarding sample-vti1
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
.
.
!
end
```

Example: Static Virtual Tunnel Interface with QoS

You can apply any QoS policy to the tunnel endpoint by including the **service-policy** statement under the tunnel interface. The following example shows how to police traffic out the tunnel interface.

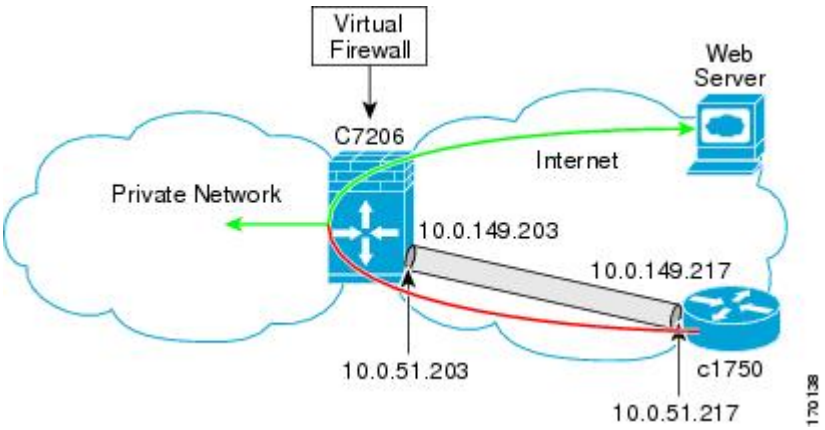
C8000 Router Configuration

```
hostname c8000
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
 service-policy output VTI
!
.
.
!
end
```

Example: Static Virtual Tunnel Interface with Virtual Firewall

Applying the virtual firewall to the SVTI tunnel allows traffic from the spoke to pass through the hub to reach the Internet. The figure below illustrates an SVTI with the spoke protected inherently by the corporate firewall.

Figure 81: Static VTI with Virtual Firewall



The basic SVTI configuration has been modified to include the virtual firewall definition:

C8000 Router Configuration

```
hostname c8000
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip nat inside
 ip inspect IOSFW1 in
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
```

```
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

Example: Dynamic Virtual Tunnel Interface Easy VPN Server

The following example illustrates the use of the DVTI Easy VPN server, which serves as an IPsec remote access aggregator. The client can be a home user running a Cisco VPN client or a Cisco IOS router configured as an Easy VPN client.

C8000 Router Configuration

```
hostname c8000
!
aaa new-model
aaa authentication login local_list local
aaa authorization network local_list local
aaa session-id common
!
ip subnet-zero
ip cef
!
username cisco password 0 cisco123
!
controller ISA 1/1
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
!
crypto isakmp client configuration group group1
 key cisco123
 pool group1pool
 save-password
!
crypto isakmp profile vpn1-ra
   match identity group group1
   client authentication list local_list
   isakmp authorization list local_list
   client configuration address respond
   virtual-template 1
!
crypto ipsec transform-set VTI-TS esp-aes esp-sha-hmac
!
crypto ipsec profile test-vti1
 set transform-set VTI-TS
!
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
```

```
!
interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip local pool group1pool 192.168.1.1 192.168.1.4
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
end
```

Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server

The following examples show that a DVTI has been configured for an Easy VPN server.

```
Router# show running-config interface Virtual-Access2

Building configuration...
Current configuration : 250 bytes
!
interface Virtual-Access2
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel source 172.18.143.246
 tunnel destination 172.18.143.208
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 no tunnel protection ipsec initiate
end
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.2.1.10 to network 0.0.0.0
     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, GigabitEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
S       192.168.1.1 [1/0] via 0.0.0.0, Virtual-Access2
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.1.0 is directly connected, GigabitEthernet0/2
S*   0.0.0.0/0 [1/0] via 172.18.143.1
```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under a Virtual Template

The following example shows how to configure VRF-aware IPsec under a virtual template to take advantage of the DVTI:

```
hostname c8000
!
ip vrf VRF-100-1
```

```
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
!
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 101
crypto isakmp profile cisco-isakmp-profile-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 102
!
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile-101
  set security-policy limit 3
  set transform-set cisco
!
crypto ipsec profile cisco-ipsec-profile-102
  set security-policy limit 5
  set transform-set Cisco
!
interface Virtual-Template101 type tunnel
  ip vrf forwarding VRF-100-1
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
  ip vrf forwarding VRF-100-2
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-102
!
```

Example: VRF-Aware IPsec with Dynamic VTI When VRF Is Configured Under a Virtual Template with the Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under a virtual template with the gateway option in an IPsec profile.

```
hostname c8000
!
ip vrf VRF-100-1
 rd 1:1
!
ip vrf VRF-100-2
 rd 1:1
!
!
!
```

```
crypto keyring cisco-100-1
 pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
 pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
 keyring cisco-100-1
 match identity address 10.1.1.0 255.255.255.0
 virtual-template 101
crypto isakmp profile cisco-isakmp-profile-100-2
 keyring cisco-100-2
 match identity address 10.1.2.0 255.255.255.0
 virtual-template 102
!
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile-101
 set security-policy limit 3
 set transform-set cisco
 set reverse-route gateway 172.16.0.1
!
crypto ipsec profile cisco-ipsec-profile-102
 set security-policy limit 5
 set transform-set cisco
 set reverse-route gateway 172.16.0.1
!
interface Virtual-Template101 type tunnel
 ip vrf forwarding VRF-100-1
 ip unnumbered Ethernet 0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
 ip vrf forwarding VRF-100-2
 ip unnumbered Ethernet 0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile cisco-ipsec-profile-102
!
```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile

```
hostname c8000
!
ip vrf VRF-100-1
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
  vrf VRF-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 1
```

```
crypto isakmp profile cisco-isakmp-profile-100-2
  vrf VRF-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 1
!
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
crypto ipsec profile cisco-ipsec-profile
  set security-policy limit 3
  set transform-set cisco
!
!
!
interface Virtual-Template 1 type tunnel
  ip unnumbered ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile
!
!
```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile and a Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under an ISAKMP profile and a gateway option in an IPsec profile:

```
hostname C8000 server
!
ip vrf VRF-100-1
 rd 1:1
!
ip vrf VRF-100-2
 rd 1:1
!
crypto keyring cisco-100-1
 pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
 pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
 vrf VRF-100-1
 keyring cisco-100-1
 match identity address 10.1.1.0 255.255.255.0
 virtual-template 1
crypto isakmp profile cisco-isakmp-profile-100-2
 vrf VRF-100-2
 keyring cisco-100-2
 match identity address 10.1.2.0 255.255.255.0
 virtual-template 1
!
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac
crypto ipsec profile cisco-ipsec-profile
 set security-policy limit 3
 set transform-set cisco
 set reverse-route gateway 172.16.0.1
!
!
```

```
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet 0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile cisco-ipsec-profile
!
!
```

Example: VRF-Aware IPsec with a Dynamic VTI When a VRF Is Configured Under Both a Virtual Template and an ISAKMP Profile

Note When separate VRFs are configured under an ISAKMP profile and a virtual template, the VRF configured under the virtual template takes precedence. This configuration is not recommended.

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI when the VRF is configured under both a virtual template and an ISAKMP profile:

```
hostname C8000 server
.
.
.
ip vrf test-vti2
 rd 1:2
 route-target export 1:1
 route-target import 1:1
!
.
.
.
ip vrf test-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
.
crypto isakmp profile cisco-isakmp-profile
 vrf test-vti2
 keyring key
 match identity address 10.1.1.0 255.255.255.0
!
.
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback 0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
.
.
```

```
.
end
```

Example: Dynamic Virtual Tunnel Interface with Virtual Firewall

The DVTI Easy VPN server can be configured behind a virtual firewall. Behind-the-firewall configuration allows users to enter the network, while the network firewall is protected from unauthorized access. The virtual firewall uses Context-Based Access Control (CBAC) and NAT applied to the Internet interface as well as to the virtual template.

```
hostname c8000
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nat inside
 ip inspect IOSFW1 in
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

Example: Dynamic Virtual Tunnel Interface with QoS

You can add QoS to the DVTI tunnel by applying the service policy to the virtual template. When the template is cloned to make the virtual access interface, the service policy will also be applied to the virtual access interface. The following example shows the basic DVTI configuration with QoS added.

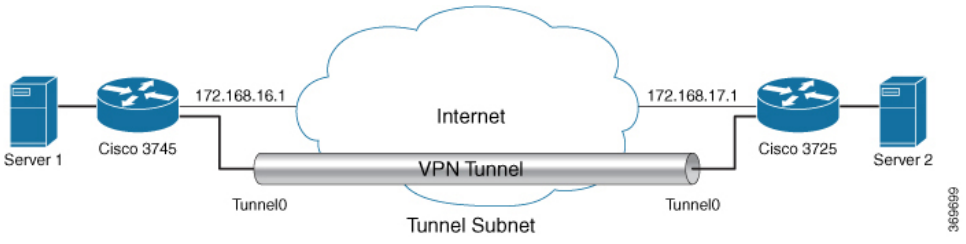
```
hostname c8000
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 service-policy output VTI
!
.
.
!
end
```

Example: Static Virtual Tunnel Interface with Multiple IPsec SAs

In the following examples an IPSec tunnel is to be established between two routers Cisco 3745 and Cisco 3725 using SVTI. The configuration uses non-any-any traffic selectors and enables the formation of multiple IPSec SAs.

Sample configuration on a Router with the IPv4 Tunnel Mode:

The following figure illustrates the reference topology for the configuration.



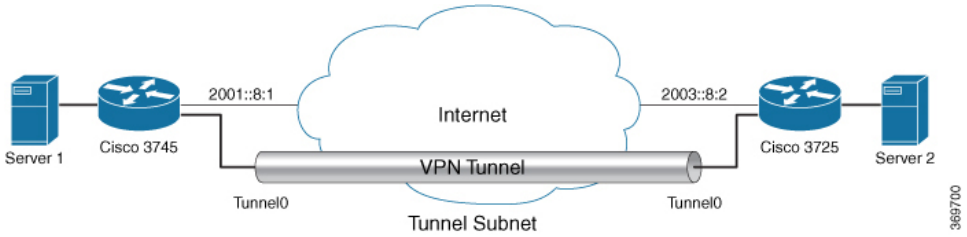
Sample configuration for the router Cisco 3745 is as follows:

```
crypto isakmp policy 1
 authentication pre-share
 group 2
!
crypto isakmp policy 5
```

```
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key example address 172.168.17.1
!
!
crypto ipsec transform-set svti1 esp-3des esp-sha-hmac
 mode tunnel
!
!
crypto ipsec profile ipsec_prof
 set transform-set svti1
!
!
!
interface Loopback0
 ip address 30.0.0.1 255.255.255.0
!
interface Loopback1
 ip address 50.0.0.1 255.255.255.0
!
interface Tunnel0
 ip address 11.1.1.2 255.255.255.0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 172.168.17.1
 tunnel protection ipsec policy ipv4 ipsec_acl1
 tunnel protection ipsec profile ipsec_prof
!
interface Ethernet0/0
 ip address 172.168.16.1 255.255.255.0
!
!
ip access-list extended ipsec_acl1
permit ip 30.0.0.0 0.0.0.255 40.0.0.0 0.0.0.255
permit ip 50.0.0.0 0.0.0.255 60.0.0.0 0.0.0.255
```

Sample configuration on a Router with the IPv6 Tunnel Mode:

The following figure illustrates the reference topology for the configuration.



Sample configuration for the router Cisco 3745 is as follows:

```
crypto isakmp policy 1
 authentication pre-share
 group 2
!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key example address ipv6 2003::8:2/112
!
!
```

```
crypto ipsec transform-set svti1 esp-3des esp-sha-hmac
 mode tunnel
!
!
crypto ipsec profile ipsec_prof
 set transform-set svti1
!
!
!
interface Loopback0
 ipv6 address 2005::10:1/112
 ipv6 enable
!
interface Loopback1
 ipv6 address 2005::15:1/112
 ipv6 enable
!
interface Loopback2
 ipv6 address 2005::20:1/112
 ipv6 enable
!
interface Tunnel0
 ip address 11.1.1.2 255.255.255.0
 ipv6 address 400::10:1/112
 ipv6 enable
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv6
 tunnel destination 2003::8:2
 tunnel protection ipsec policy ipv6 ipsec_acl2
 tunnel protection ipsec profile ipsec_prof
!
interface Ethernet0/0
 ipv6 address 2001::8:1/112
 ipv6 enable
!
!
ipv6 access-list ipsec_acl2
sequence 10 permit ipv6 host 2005::10:1 host 2005::11:1
sequence 20 permit ipv6 host 2005::15:1 host 2005::16:1
sequence 30 permit ipv6 host 2005::20:1 host 2005::21:1
```

Example: Configuring Tunnel Mode as Dual-overlay

The following example shows how to configure tunnel mode as dual-overlay:

```
Device# configure terminal
Router(config)# interface tunnel 1
Router(config-if)# ipv6 enable
Router(config-if)# tunnel source ethernet 0/0
Router(config-if)# tunnel mode ipsec dual-overlay
Router(config-if)# tunnel destination 89.89.89.1 255.255.255.255.0
Device(config-if)# tunnel protection IPsec profile ipsecprof
```

Verifying the Tunnel Mode as Dual-overlay Configuration

Use the following commands to troubleshoot your configuration:

- **Show crypto session [detail]**
- **Show crypto ipsec sa**
- **Show crypto map**

- **Show crypto socket**
- **Show crypto ikev2 session [detail]**

```
Device# show crypto map
Crypto Map: "Tunnel0-head-0" IKEv2 profile: prof

Crypto Map IPv4 "Tunnel0-head-0" 65536 ipsec-isakmp
 IKEv2 Profile: prof
 Profile name: prof
 Security association lifetime: 4608000 kilobytes/120 seconds
 Dualstack (Y/N): N

 Responder-Only (Y/N): N
 PFS (Y/N): N
 Mixed-mode : Disabled
 Transform sets={
  default:  { esp-aes esp-sha-hmac  } ,
 }

Crypto Map IPv4 "Tunnel0-head-0" 65537 ipsec-isakmp
 Map is a PROFILE INSTANCE.
 Peer = 10.10.10.2
 IKEv2 Profile: prof
 Extended IP access list
     access-list  permit ip any any
 Current peer: 10.10.10.2
 Security association lifetime: 4608000 kilobytes/120 seconds
        Dualstack (Y/N): Y
        TRUE  ident (addr/mask/prot/port): {LOCAL -> REMOTE}
             0.0.0.0/0.0.0.0/0/0 -> 0.0.0.0/0.0.0.0/0/0
             ::/0.0.0.0/0/0 -> ::/0/0/0
 Responder-Only (Y/N): N
 PFS (Y/N): N
 Mixed-mode : Disabled
 Transform sets={
  default:  { esp-aes esp-sha-hmac  } ,
 }
 Always create SAs
 Interfaces using crypto map Tunnel0-head-0:
  Tunnel0

Device# show  crypto ipsec sa

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 10.10.10.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
    TRUE  ident (addr/mask/prot/port): {LOCAL -> REMOTE}
             0.0.0.0/0.0.0.0/0/0 -> 0.0.0.0/0.0.0.0/0/0
             ::/0.0.0.0/0/0 -> ::/0/0/0
   current_peer 10.10.10.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.2
     plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
```

```
     current outbound spi: 0x4776A36B(1198957419)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0xA97EDEE7(2843664103)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 4, flow_id: 4, sibling_flags FFFFFFFF80000040, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4377587/76)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x4776A36B(1198957419)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 3, flow_id: 3, sibling_flags FFFFFFFF80000040, crypto map: Tunnel0-head-0
         sa timing: remaining key lifetime (k/sec): (4377587/76)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:

     outbound pcp sas:
```

Device# show crypto socket

```
Number of Crypto Socket connections 1

   Tu0 Peers (local/remote): 10.10.10.1/10.10.10.2
       Local Ident  (addr/mask/port/prot): (0.0.0.0/0.0.0.0/0/0)
       Remote Ident (addr/mask/port/prot): (0.0.0.0/0.0.0.0/0/0)
        TRUE  ident (addr/mask/prot/port): {LOCAL -> REMOTE}
             0.0.0.0/0.0.0.0/0/0 -> 0.0.0.0/0.0.0.0/0/0
             ::/0.0.0.0/0/0 -> ::/0/0/0
       IPSec Profile: "prof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "prof" Map-name: "Tunnel0-head-0"
```

Device# **show cry ikev2 session**

```
 IPv4 Crypto IKEv2 Session

Session-id:1, Status:UP-ACTIVE, IKE count:1, CHILD count:1

Tunnel-id Local                 Remote                fvrf/ivrf            Status
1         10.10.10.1/500        10.10.10.2/500        none/none            READY
      Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: PSK,
Auth verify: PSK
      Life/Active Time: 86400/145 sec
      CE id: 1001, Session-id: 1
      Local spi: 25A0B173944015D3       Remote spi: 9F0C7677425670E1
Child sa:
          local selector  0.0.0.0/0 - 255.255.255.255/65535
          local selector  ::/0 - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF/65535
          remote selector 0.0.0.0/0 - 255.255.255.255/65535
          remote selector ::/0 - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF/65535
          ESP spi in/out: 0xA97EDEE7/0x4776A36B
```

```
 IPv6 Crypto IKEv2 Session

Device# show crypto session
Crypto session current status

Interface: Tunnel0
Profile: prof
Session status: UP-ACTIVE
Peer: 10.10.10.2 port 500
  Session ID: 1
  IKEv2 SA: local 10.10.10.1/500 remote 10.10.10.2/500 Active
  IPSEC FLOW: permit ip   0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
  TRUE IDENT (addr/mask/prot/port): {LOCAL -> REMOTE}
             0.0.0.0/0.0.0.0/0/0 -> 0.0.0.0/0.0.0.0/0/0
             ::/0.0.0.0/0/0 -> ::/0/0/0
        Active SAs: 2, origin: crypto map
```

Additional References for IPsec Virtual Tunnel Interface

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	*Configuring Security for VPNs with IPsec*
QoS configuration	*Cisco IOS Quality of Service Solutions Configuration Guide*
EasyVPN configuration	• *Cisco Easy VPN Remote* • *Easy VPN Server*
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*

Standard/RFC	Title
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*The Internet Key Exchange (IKE)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec Virtual Tunnel Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 211: Feature Information for IPsec Virtual Tunnel Interfaces

Feature Name	Releases	Feature Configuration Information
Dynamic IPsec VTIs	12.3(7)T 12.3(14)T	Dynamic VTIs enable efficient use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs. The dynamic VTI simplifies VRF-aware IPsec deployment. The following commands were introduced or modified: **crypto isakmp profile, interface virtual-template, show vtemplate, tunnel mode, virtual-template.**

Feature Name	Releases	Feature Configuration Information
FlexVPN Mixed Mode Support	15.4(2)T Cisco IOS XE Release 3.10S	The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic. This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.
Multi-SA for Dynamic VTIs	15.2(1)T Cisco IOS XE Release 3.2S	The DVTI can accept multiple IPsec selectors that are proposed by the initiator. The following commands were introduced or modified: **set security-policy limit, set reverse-route.**
Static IPsec VTIs	12.2(33)SRA 12.2(33)SXH 12.3(7)T 12.3(14)T Cisco IOS XE Release 2.1	IPsec VTIs provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.
Tunnel Mode Auto Selection	15.4(2)T Cisco IOS XE Release 3.12S	The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. The following command was introduced or modified: **virtual-template**

Feature Name	Releases	Feature Configuration Information
FlexVPN Mixed Mode v6 over v4 Transport	Cisco IOS XE Everest 16.4.1	The FlexVPN Mixed Mode v6 over v4 Transport feature provides support for carrying IPv6 traffic over IPsec IPv4 transport. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.
IPsec Dual Stack Support on Non Cisco Devices	Cisco IOS XE Cupertino 17.9.x	This feature provides the capabilities to carry both IPv4 and IPv6 traffic using a single IPsec Security Association (SA) that is tunnelled over IPv4. From IOS XE release 17.9.1a onwards, Cisco supports specific subnets in the access control list when the ingress end of the tunnel interface is configured with a third party IPsec client. With the introduction of the SVTI single security association dual stack feature, you can now manage the business-to-business services and other IOT business efficiently.

CHAPTER 158

Session Initiation Protocol Triggered VPN

Session Initiation Protocol Triggered VPN (SIP-Triggered VPN or VPN-SIP) is a service offered by service providers where a VPN is set up using Session Initiation Protocol (SIP) for on-demand media or application sharing between peers. The VPN-SIP feature defines the process in which two SIP user agents resolve each other's IP addresses, exchange the fingerprints of their self-signed certificates, third-party certificates, or pre-shared key securely, and agree to establish an IPsec-based VPN.

Service providers offer the VPN-SIP service to their customers that have SIP-based services such as bank ATMs or branches. This VPN-SIP service replaces an ISDN connection for backup network functionality. If the primary broadband service link goes down, these bank ATMs or branches connect to their central headend or data centres through the VPN-SIP service.

The SIP server of the service provider, which coordinates the VPN-SIP service, is also used for billing of the service based on the time the service is used.

Feature Information for VPN-SIP

Table 212: Feature Information for VPN-SIP

Feature Name	Releases	Feature Information
Session Initiation Protocol Triggered VPN		VPN-SIP is a service offered by service providers where a VPN is setup for on-demand media or application sharing between peers, using Session Initiation Protocol (SIP). The following commands were introduced: **nat force-encap, show vpn-sip session, show vpn-sip sip, show vpn-sip registration-status, vpn-sip local-number, vpn-sip logging, vpn-sip tunnel source**.

Information about VPN-SIP

Components for VPN-SIP Solution

VPN-SIP uses IPSec Static Virtual Tunnel Interface (SVTI). IPSec SVTI stays in active (UP) state even when there is no IPSec security association (SA) established between the tunnel interface and the SVTI peer.

The following are three components for the VPN-SIP Solution:

- SIP
- VPN-SIP
- Crypto (IP Security (IPsec), Internet Key Exchange (IKE), Tunnel Protection (TP), Public Key Infrastructure (PKI) modules within crypto)

Sesssion Initiation Protocol

SIP is used as a name resolution mechanism to initiate an IKE session. VPN-SIP uses SIP service to establish a VPN connection to a home or a small business router that does not have a fixed IP address. This connection is achieved using self-signed certificates or pre-shared keys. SIP negotiates the use of IKE for media sessions in the Session Description Protocol (SDP) offer-and-answer model.

SIP is statically configured. One tunnel interface must be configured for each remote SIP number.

SIP also provides billing capabilities for service providers to charge customers based on the SIP number, for using the VPN-SIP service. Billing based on SIP numbers happens in the service provider network and is independent of the end devices like Cisco VPN-SIP routers.

VPN-SIP Solution

VPN-SIP is the central block that coordinates between SIP and Crypto modules, and provides an abstraction between them.

When traffic destined to a remote network behind a SIP number is routed to the tunnel interface, the IPSec control plane gets a trigger from packet switching path as there is no IPSEC SA configured to that peer. IPsec control plane passes the trigger to VPN-SIP as the tunnel is configured for VPN-SIP.

Note Static routes for remote networks for that SIP number must be configured to point to that tunnel interface.

When the VPN-SIP service is triggered, SIP sets up the call with a SIP phone number pair. SIP also passes incoming call details to the VPN-SIP and negotiates IKE media sessions using local address and fingerprint information of the local self-signed certificate or pre-shared key. SIP also passes remote address and fingerprint information to VPN-SIP.

The VPN-SIP service listens to tunnel status updates and invokes SIP to tear down the SIP session. The VPN-SIP service also provides a means to display current and active sessions.

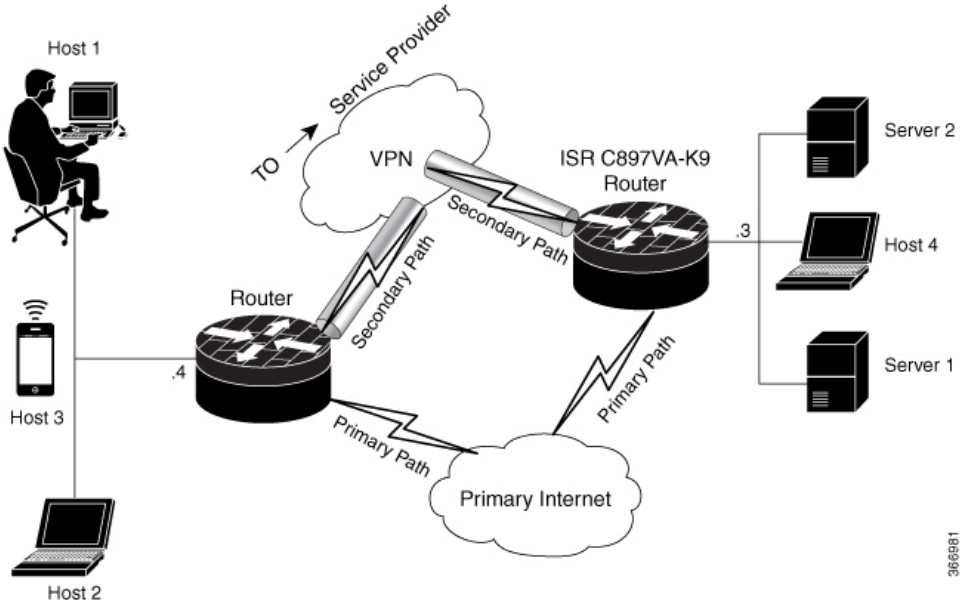
Feature at a glance

The following steps summarize how the VPN-SIP feature works:

- IP SLA monitors the primary link using route tracking. When the primary link fails IP SLA detects this failure.
- Once the primary path fails, IP SLA switches the default route to the higher metric route that is configured on the router.
- When relevant traffic tries to flow using the secondary link, SIP sends an invite message to the SIP server to obtain the VPN peer information.
- The router receives the VPN peer information (IP address, local and remote SIP numbers, IKE port, and finger print) and it establishes VPN-SIP tunnel.
- When the primary path comes back up, IP SLA detects the primary path and the route falls back to the original path. When the idle timer expires, IPSec is torn down and a SIP call is disconnected.

Following is the topology for the VPN-SIP solution:

Figure 82: VPN-SIP Topology



SIP Call Flow

The SIP call flow is divided into initiation at the local peer and call receipt at the remote peer.

At SIP Call Intitiation

When packets are routed to an SVTI interface in data plane, the SIP call must be placed to the peer SIP number to resolve its address, so that VPN tunnel can be brought up.

- When local auth-type is PSK, IKEv2 finds the matching key for a peer SIP number. The IKEv2 keyring must be configured with id_key_id type (string) as SIP number for each SIP peer. IKEv2 computes the fingerprint of the looked-up key and passes it to VPN-SIP.
- When local auth-type is a self-signed certificate or an third-party certificate, IKEv2 computes the fingerprint of the local certificate configured under the IKEv2 profile and passes it to the VPN-SIP

The VPN-SIP module interacts with SIP to setup SIP call to the peer. When the call is successful, VPN-SIP sets the tunnel destination of SVTI to the resolved IP address, requesting SVTI to initiate the VPN tunnel.

Note When a wildcard key is required, use the authentication local pre-share key command and the authentication remote pre-share key command in IKEv2 profile.

When SIP call is received at the remote peer

When a SIP call is received from a peer, following interactions occur between various crypto modules:

- The Tunnel Protection helps VPN-SIP module to set tunnel destination address.

- IKEv2 returns local auth-type (PSK or PKI) and local fingerprint to the VPN-SIP module. When local auth-type is PSK, IKEv2 finds a matching key for a corresponding SIP number.

Note IKEv2 only knows peer by its SIP number.

During the SIP call negotiation between peers, each peer must select a unique local IKEv2 port number to be exchanged over the SDP. To support different port numbers for each session, the VPN-SIP module programmatically configures IP Port Address Translation (PAT) to translate between IKEv2 port (4500) and the port number exchanged over SDP. For the translation to work IP NAT must be configured on secondary link and the loopback interface configured as the VPN-SIP tunnel source. The lifetime of the translation is limited to the lifetime of the VPN-SIP session.

SDP Offer and Answer

Following is the sample for SDP offer and answer that is negotiated in the SIP call as defined in RFC 6193:

```
offer SDP
      ...
      m=application 50001 udp ike-esp-udpencap
      c=IN IP4 10.6.6.49
      a=ike-setup:active
      a=fingerprint:SHA-1 \
      b=AS:512
      4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
      ...

   answer SDP
      ...
      m=application 50002 udp ike-esp-udpencap
      c=IN IP4 10.6.6.50
      a=ike-setup:passive
      a=fingerprint:SHA-1 \
      b=AS:512
      D2:9F:6F:1E:CD:D3:09:E8:70:65:1A:51:7C:9D:30:4F:21:E4:4A:8E
```

As part of the SDP negotiation, both peers negotiate the maximum bandwidth rate for the VPN-SIP session using the b=AS :number SDP attribute. if the peers mention different bandwidth numbers in their SDP, both of them should honor the minimum value as the maximum bandwidth. If b=AS :number SDP attribute is missing in the offer or answer, the SIP call is not successfully set up.

The negotiated maximum bandwidth is applied on the SVTI tunnel interface through the programmatically configured QoS policy in the output direction. The programmatically configured QoS policy is not applied and session fails, if there is a pre-existing statically configured policy.

Once SIP call is complete and address of the peer is resolved, VPN-SIP sets tunnel destination of SVTI and sends a request to initiate tunnel.

IKEv2 Negotiation

Following is the process for IKEv2 Security Session (SA) negotiation:

- Before starting the session, IKEv2 checks with VPN-SIP if the session is a VPN-SIP session.
- If it's a VPN-SIP session and local auth-type is PSK, IKEv2 looks up the PSK key pair using SIP number of the peer instead of IP address of the peer.

- For validating self-signed certificate, IKEv2 checks if the certificate is self-signed and validates the certificate.
    - In addition to existing AUTH payload validation as part of IKEv2 protocol, IKEv2 calculates hash of the received certificate or looked-up PSK and compares with the fingerprint from SIP negotiation that IKEv2 queries from VPN-SIP module. Only if the fingerprint matches, IKEv2 considers authentication of peer is valid. If not, IKEv2 declares that peer has failed to authenticate and fails the VPN session.

VPN-SIP solution depends on IPSEC idle timer to detect that traffic is no longer routed over the backup VPN. The idle-time configuration under the IPSec Profile is mandatory for session to be disconnected when there is no traffic. 120 seconds is the recommended time.

VPN-SIP and SIP coordinate to tear down SIP call.

When IPsec idle time expires the VPN-SIP module informs the IKEv2 to bring down the IPsec tunnel. VPN-SIP requests the SIP module to disconnect the SIP call, without waiting for confirmation from the IKEv2.

When SIP call disconnect is received from the peer, VPN-SIP module informs the IKEv2 to bring down the IPsec tunnel, and acknowledges to SIP to tear down the SIP call.

Prerequisites for VPN-SIP

- Security K9 license must be enabled on the router.
- The routers must have a minimum memory of 1 GB.
- For the SIP register request of the SIP User Agent to succeed, the SIP registrar must be available to the VPN-SIP routers.
- The DHCP server must support option 120 and 125 to obtain the SIP server address, which is needed for registration and establishing the SIP session.
- Proper routing configurations must be completed to ensure backup WAN path is used when primary path is down.
- Maximum Transmission Unit (MTU) of the tunnel interface must be less than the MTU of the secondary WAN interface.
- When self-signed or third-party certificates are used for IKEv2 authentication, configure IKEv2 fragmentation on the VPN-SIP router to avoid fragmentation at the IP layer.
- NAT SIP ALG must be disabled.
- Caller ID notification service must be configured in the network.

Restrictions for VPN-SIP

- VPN-SIP and CUBE/SIP gateway cannot be configured on the same device. When CUBE license is active on the device, only CUBE will be functional.
- Only IPv4 is supported for transport and media (IPv4 transport for SIP registration, SIP signaling, and IPv4 packets encrypted over IPv4 transport).

- SIP signalling with peer devices behind NAT is not supported (ICE and STUN are not supported.
- SIP negotiation is supported only in global VRF.
- Remote-access VPN features like private address assignment, configuration mode exchange (CP payloads), routes exchange, are not supported.
- Routing protocols over the VPN-SIP session are not supported.
- Only Rivest-Shamir-Addleman (RSA) server self-signed certificates are supported.
- Pre-shared key lookup functionality using authentication, authorization, and accounting (AAA) is not supported.
- The IPSec idle timer is configured per IPSec profile using the ipsec-profile command. The idle time is the same for all VPN-SIP sessions that use a specific IPSec profile.
- Track objects that are used for IPSLA monitoring, have a maximum limit of 1000 objects in Cisco IOS software. When one track object is used to track one peer router, maximum number of VPN-SIP sessions that one IOS device can have is limited by the maximum number of track objects.
- Only one local SIP number is supported on Cisco IOS software.
- If there is a pre-existing statically configured policy, the programmatically configured QoS policy is not applied and session fails. Remove any statically configured QoS policy on the SVTI interface.
- Cisco does not support the interoperability with VPN-SIP implementation of other vendors.
- For the class policies included in the `policy-map` attached to the VPN-SIP tunnel, only Priority Queueing and Class-Based Weighted Fair Queueing (CBWFQ) are supported.
- For CBWFQ configurations, only the `bandwidth percent` *`percent`* command is supported. The `bandwidth` *`bandwidth`* command is not supported as the bandwidth of the VPN-SIP session varies depending on the negotiation with the peer router.
- VPN-SIP configuration is not supported on IPv6.
- VPN-SIP configuration is supported only in autonomous mode.
- Complex SIP call scenarios such as refer, fork etc. are not supported in VPN-SIP configuration.

How to Configure VPN-SIP

Configuring VPN-SIP

The following steps describe the process of configuring VPN-SIP:

1. Configure the tunnel authentication using third party certificates, self-signed certificates, or pre-shared keys.
   a. Tunnel Authentication using Certificates

      Configure a trustpoint to obtain a certificate from a certification authority (CA) server that is located in the customer's network. This is required for tunnel authentication. Use the following configuration:

```
peer1(config)# crypto pki trustpoint CA
 enrollment url http://10.45.18.132/
 serial-number none
 subject-name CN=peer2
 revocation-check crl
 rsakeypair peer2

peer2(config)# crypto pki authenticate CA
Certificate has the following attributes:
       Fingerprint MD5: F38A9B4C 2D80490C F8E7581B BABE7CBD
      Fingerprint SHA1: 4907CC36 B1957258 5DFE23B2 649E7DDA 99BDB7C3
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

peer2(config)#crypto pki enroll CA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: CN=peer2
% The subject name in the certificate will include: peer2
% Include an IP address in the subject name? [no]:
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA' command will show the fingerprint.
Certificate map for Trustpoint
crypto pki certificate map data 1
issuer-name co cn = orange
```

b. Tunnel authentication using self-signed certificate

Configure a PKI trust point to generate a self-signed certificate on the device, when authenticating using a self-signed certificate. Use the following configuration:

```
peer4(config)#crypto pki trustpoint Self
    enrollment selfsigned
    revocation-check none
    rsakeypair myRSA
    exit
crypto pki enroll Self

Do you want to continue generating a new Self Signed Certificate? [yes/no]: yes
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created
```

c. Configure tunnel authentication using a pre-shared key

```
crypto ikev2 keyring keys
peer peer1
identity key-id 1234
pre-shared-key key123
```

2. **a.** Configure IKEv2 Profile for Certificate

```
crypto ikev2 profile IPROF
match certificate data
identity local key-id 5678
authentication remote rsa-sig
```

```
authentication local rsa-sig
keyring local keys
pki trustpoint self
nat force-encap
```

b. Configure an IKEv2 Profile for pre-shared keys

```
crypto ikev2 profile IPROF
match identity remote any
identity local key-id 5678
authentication remote pre-share
authentication local pre-share
keyring local keys
nat force-encap
```

Note To complete the IKEv2 SA configuration, the **nat force-encap** command must be configured on both peers. Since, UDP encapsulation is negotiated in SDP, IKEv2 must start and continue on port 4500.

3. Configure an IPsec profile

```
crypto ipsec profile IPROF
set security-association idle-time 2000
```

4. Configure a LAN side interface

```
interface Vlan101
         ip address 10.3.3.3 255.255.255.0
         no shutdown
!
     interface GigabitEthernet2
         switchport access vlan 101
         no ip address
```

5. Configure a loopback interface

The loopback interface is used as the source interface for the secondary VPN tunnel.

```
interface loopback 1
    ip address 10.11.1.1 255.0.0.0
    ip nat inside
```

6. Configure a secondary interface.

Note Make sure the secondary interface is configured to receive the IP address, SIP server address, and vendor specific information via DHCP.

```
interface GigabitEthernet8
     ip dhcp client request sip-server-address
     ip dhcp client request vendor-identifying-specific
     ip address dhcp
    ip nat outside
```

7. Configure the tunnel interface

```
interface Tunnel1
     ip address 10.3.2.1 255.255.255.255
     load-interval 30
     tunnel source Loopback1
```

```
tunnel mode ipsec ipv4
tunnel destination dynamic
tunnel protection ipsec profile IPROF ikev2-profile IPROF
vpn-sip local-number 5678 remote-number 1234 bandwidth 1000
```

Use the **vpn-sip local-number** *local-number* **remote-number** *remote-number* **bandwidth** *bw-number* command to configure the sVTI interface for VPN-SIP. Bandwidth is the maximum data transmission rate that must be negotiated with this peer and the negotiated value is set on the tunnel interface. Allowed values are 64, 512, and 1000 kbps.

Once an SVTI is configured for VPN-SIP, changes cannot be made to tunnel mode, tunnel destination, tunnel source, and tunnel protection. To change the mode, source, destination, or tunnel protection you must remove the VPN-SIP configuration from the SVTI interface.

8. Add static routes to destination networks

Add a secondary route with a higher metric.

```
ip route 192.168.10.0 255.255.255.0 Tunnel0 track 1
ip route 192.168.10.0 255.255.255.0 Tunnel1 254
```

9. Configure IP SLA

```
ip sla 1
         icmp-echo 10.11.11.1
         threshold 500
         timeout 500
         frequency 2
        ip sla schedule 1 life forever start-time now
```

10. Configure route tracking

```
track 1 ip sla 1 reachability
```

11. Enable VPN-SIP

```
vpn-sip enable
vpn-sip local-number 5678 address ipv4 GigabitEthernet8
vpn-sip tunnel source Loopback1
vpn-sip logging
```

To configure VPN-SIP, you must configure local SIP number and local address. The **vpn-sip local-number** *SIP-number* **address ipv4** *WAN-interface-name* command configures the local SIP number that is used for SIP call and the associated IPv4 address.

Note Only IPv4 addresses can be configured. Crypto module does not support dual stack.

- Backup WAN interface address may change based on DHCP assignment.

When the primary WAN interface is functional, the destination of the VPN-SIP tunnel is set to the backup WAN interface, so that the tunnel interface is active. Destination is set to IP address of the peer that is learnt from SDP of SIP negotiation when traffic is routed to the tunnel interface. When primary WAN interface fails and the back routes are activated, packets are routed to the sVTI through backup.

Note We recommend that you use an unused non-routable address as the address of the loopback interface and do not configure this loopback interface for any other purpose. Once a loopback interface is configured, VPN-SIP listens to any updates to the interface and blocks them. The **vpn-sip logging** command enables the system logging of VPN-SIP module for events, such as session up, down, or failure.

Verifying VPN-SIP on a Local Router

Verifying Registration Status

```
Peer1# show vpn-sip registration-status
 SIP registration of local number 0388881001 : registered 10.6.6.50
```

Verifying SIP Registrar

```
Peer1#show vpn-sip sip registrar

Line          destination      expires(sec)  contact   transport     call-id
=============================================================================================================
0388881001    example.com         2359        10.6.6.50  UDP
3176F988-9EAA11E7-8002AFA0-8EF41435
```

Verifying VPN-SIP Status

```
Peer1#show vpn-sip session detail
VPN-SIP session current status

Interface: Tunnel1
   Session status: SESSION_UP (I)
   Uptime        : 00:00:42
   Remote number : 0388881001 =====> This is the Remote Router's SIP number
   Local number  : 0388882001 =====> Local router's SIP number
   Remote address:port: 10.6.6.49:50002
   Local address:port : 10.6.6.50:50001
   Crypto conn handle: 0x8000017D
   SIP Handle        : 0x800000C7
   SIP callID        : 1554
   Configured/Negotiated bandwidth: 64/64 kbps
```

Verifying Crypto Session

```
Peer1# show crypto session detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect, U - IKE Dynamic Route Update
S - SIP Vpn-sip

Interface: Tunnel1
Profile: IPROF
Uptime: 00:03:53
Session status: UP-ACTIVE
Peer: 10.6.6.49 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 10.6.6.49
      Desc: (none)
```

```
  Session ID: 43
  IKEv2 SA: local 10.11.1.1/4500 remote 10.6.6.49/50002 Active
          Capabilities:S connid:1 lifetime:23:56:07 ====> Capabilities:S indicates this is
 a  SIP VPN_SIP Session
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 6 drop 0 life (KB/Sec) 4222536/3366
        Outbound: #pkts enc'ed 4 drop 0 life (KB/Sec) 4222537/3366
```

Verifying IP NAT Translations

```
Peer1#sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 2.2.2.2:4500       10.6.6.50:50001    10.6.6.49:50002    10.6.6.49:50002
```

Verifying DHCP SIP Configuration

```
Peer9#show vpn-sip sip dhcp
SIP DHCP Info

SIP-DHCP interface:  GigabitEthernet8

SIP server address:
Domain name:         dns:example.com
```

Configuration Examples for VPN-SIP

Using self-signed certificates for authentication

The following is sample configuration to configure VPN-SIP using self-signed certificates for authentication. There is no distinction between initiator and responder role in VPN-SIP. The configuration on a peer node will be identical with local SIP numbers changed.

```
// Self-signed certificate
crypto pki trustpoint selfCert
  rsakeypair myRSA
  enrollment selfsigned
  revocation-check none
!
crypto ikev2 profile vpn-sip-profile
 match identity remote any
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint selfCert // Use same self-signed trustpoint for sign and verify
 nat force-encap
!
crypto ipsec profile vpn-sip-ipsec
 set security-association idle-time 120
!
vpn-sip enable
vpn-sip local-number 0388883001 address ipv4 GigabitEthernet1
vpn-sip tunnel source Loopback11
vpn-sip logging
!
// one tunnel per peer - configuration is for peer with a SIP-number of 0388884001
int tunnel0
 ip unnumbered loopback 0
 tunnel source loopback11
 tunnel mode ipsec ipv4
```

```
 tunnel destination dynamic
 tunnel protection ipsec profile vpn-sip-ipsec ikev2-profile vpn-sip-profile
 vpn-sip local-number 0388883001 remote-number 0388884001 bandwidth 1000
!
// ip unnumbered of tunnel interfaces
int loopback 0
  ip address 10.21.1.1 255.255.255.255
!
int loopback11
ip address 10.9.9.9 255.255.255.255
ip nat inside
!
// one tunnel per peer - this is for peer with SIP-number 0388885001
int tunnel1
 ip unnumbered loopback 0
 tunnel source loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile vpn-sip-ipsec ikev2-profile iprof
 vpn-sip sip-local 0388883001 sip-remote 0388885001 bandwidth 1000
!
interface GigabitEthernet8
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp
ip nat outside


// backup routes configured with higher AD so that these routes will be activated only when
 primary path goes down. AD need to be chosen to be greater than that of primary route.
ip route 10.0.0.0 255.0.0.0 tunnel 0 250
ip route 10.1.0.0 255.0.0.0 tunnel 0 250
ip route 10.2.0.0 255.0.0.0 tunnel 0 250
ip route 10.3.0.0 255.0.0.0 tunnel 0 250
```

Configuring DHCP in VPN-SIP

Configure DHCP for VPN-SIP

From Cisco IOS XE Release 17.11.1a, you can install a VPN-SIP-enabled router behind a home gateway (HGW). In this installation, the HGW assigns an extension number to the tunnel interface through the Dynamic Host Configuration Protocol (DHCP) instead of a fixed telephone number. This allows you to aggregate data and voice on your network, which can be useful in scenarios where you need to share the same physical subscriber line for both analog and digital data.

In addition, to comply with the HGW network specifications, DHCP for VPN-SIP requires the MAC address of the WAN-side interface to the HGW network through the vendor-class-data DHCP option. With this configuration, the device communicates the MAC address of its own WAN interface to the home gateway network through the vendor-class-data option of the DHCP requests.

Supported PIDs and Firmware

The following table specifies the HGW PIDs and the firmware versions that are tested. Cisco does not provide support for the HGW installed at a customer's location or the operation of an HGW. We recommend that you verify your environment before using this feature.

HGW PID	Firmware Version
RT-400NE	8.06
RT-400MI	09.00.0015
RT-400KI	08.00.0040
RT-500MI	08.00.0004
RT-500KI	08.00.0020
RX-600MI	01.00.0001
RX-600KI	01.00.0001
OG410Xi	2.32
OG410Xa	2.32

Configure DHCP for VPN-SIP

When you configure a DHCP local number, the device defers SIP registration until it receives a DHCP response. The device expects the DHCP server to provide an extension number. This extension number is then used to register with the SIP server. On successful registration, the device initiates a session with the SIP server and receives an extension number, an external number, and other available numbers through a 200 OK response.

Note The external number is the number with which the router is identified globally. This external number is also required to establish a data connection.

With the DHCP enhancement, there are two channels for data connection—SIP signalling channel and IPsec data connection. If the data packets require tunnel protection, a SIP call is initiated.

Perform the following procedures to configure DHCP for VPN-SIP.

Enable the DHCP Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type number**
4. **ip dhcp client request sip-server-address**
5. **ip dhcp client request vendor-identifying-specific**
6. **ip address dhcp**
7. **ip dhcp client vendor-class mac-address**
8. **ip nat outside**

9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface type number** **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip dhcp client request sip-server-address** **Example:** `Router(config-if)# ip dhcp client request sip-server-address`	Configures the DHCP client to request a SIP server address from a DHCP server.
Step 5	**ip dhcp client request vendor-identifying-specific** **Example:** `Router(config-if)# ip dhcp client request vendor-identifying-specific`	Configures the DHCP client to request vendor-specific information from a DHCP server.
Step 6	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Acquires an IP address on the interface from the DHCP.
Step 7	**ip dhcp client vendor-class mac-address** **Example:** `Router(config-if)# ip dhcp client vendor-class mac-address`	Complies with the HGW's DHCP specification.
Step 8	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 9	**exit** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enable DHCP Client Sample Configuration

The following is a sample code for enabling a DHCP client.

```
interface GigabitEthernet 0/0/0
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp
 ip dhcp client vendor-class mac-address
 ip nat outside
```

Configure Tunnel Authentication

You can configure tunnel authentication by using third-party certificates, self-signed certificates, or by using preshared keys (PSKs). To configure tunnel authentication, perform one of the following tasks.

Configure Tunnel Authentication Using Certificates

Configure a trustpoint to obtain a certificate from a certification authority (CA) server that is located in the customer's network. This is required for tunnel authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** name
4. **enrollment url** `url`
5. **serial-number**
6. **subject-name** `[subject-name]`
7. **revocation-check crl**
8. **rsakeypair**
9. **crypto pki authenticate CA**
10. **crypto pki enroll CA** `name`
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** name **Example:** `Router(config)# crypto pki trustpoint CA`	Specifies the trustpoint and a given name, and enters the ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	**enrollment url `url`** **Example:** `Router(ca-trustpoint)# enrollment url http://10.45.18.132/`	Specifies the URL of the CA to which your router should send certificate requests.
Step 5	**serial-number** **Example:** `Router(ca-trustpoint)# serial-number`	Specifies the serial number of the router in the certificate request, unless you use the **none** keyword. Use the **none** keyword if you don't want to include a serial number in the certificate request.
Step 6	**subject-name `[subject-name]`** **Example:** `Router(ca-trustpoint)# subject-name CN=peer2`	Specifies the requested subject name that is used in the certificate request. If you don’t specify the subject name, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 7	**revocation-check crl** **Example:** `Router(ca-trustpoint)# revocation-check crl`	Checks the validity of the certificate through the Certificate Revocation Lists (CRL) mechanism.
Step 8	**rsakeypair** **Example:** `Router (ca-trustoint)# rsakeypair peer2`	Provides a key pair for the trustpoint.
Step 9	**crypto pki authenticate CA** **Example:** `Router(config)# crypto pki authenticate CA`	Authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.
Step 10	**crypto pki enroll CA `name`** **Example:** `Router(config)# crypto pki enroll CA`	Generates the certificate request and displays the request for copying and pasting into the certificate server. You are prompted for enrollment information such as whether to include the router FQDN and IP address in the certificate request. You are also given a choice about displaying the certificate request on the console terminal.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configuring Tunnel Authentication Using Certificates

This is the sample code for configuring tunnel authentication using certificates.

```
peer1(config)# crypto pki trustpoint CA
 enrollment url http://10.45.18.132/
 serial-number none
 subject-name CN=peer2
 revocation-check crl
 rsakeypair peer2
```

```
peer2(config)# crypto pki authenticate CA
Certificate has the following attributes:
Fingerprint MD5: F38A9B4C 2D80490C F8E7581B BABE7CBD
Fingerprint SHA1: 4907CC36 B1957258 5DFE23B2 649E7DDA 99BDB7C3
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

peer2(config)#crypto pki enroll CA
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration. Please make a
note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: CN=peer2
% The subject name in the certificate will include: peer2
% Include an IP address in the subject name? [no]:
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA' command will show the fingerprint.
Certificate map for Trustpoint
crypto pki certificate map data 1
issuer-name co cn = orange
```

Configure Tunnel Authentication Using Self-Signed Certificates

To configure tunnel authentication using a self-signed certificate, run the **crypto pki trustpoint self** command. This command enables you to configure a PKI trust point to generate a self-signed certificate on the device.

```
Router(config)# crypto pki trustpoint self
enrollment self signed
revocation-check none
rsakeypair myRSA
exit

crypto pki enroll self
Do you want to continue generating a new Self Signed Certificate? [yes/no]: yes
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Configure Tunnel Authentication Using PreShared Keys

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 keyring keyring-name**
4. **peer name**
5. **address {ipv4-address [mask] | ipv6-address prefix}**
6. **identity {address { ipv4-address | ipv6-address} | fqdn name | email email-id | key-id key-id}**
7. **pre-shared-key {local| remote} {0| 6| line}**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 keyring keyring-name** **Example:** `Router(config)# crypto ikev2 keyring kyr1`	Defines an IKEv2 keyring, and enters IKEv2 keyring configuration mode.
Step 4	**peer name** **Example:** `Router(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group, and enters IKEv2 keyring peer configuration mode.
Step 5	**address {ipv4-address [mask] | ipv6-address prefix}** **Example:** `Router(config-ikev2-keyring-peer)# address 10.0.0.1 255.255.255.0`	Specifies an IP address or a range for the peer. This IP address is the IKE endpoint address and is independent of the identity address.
Step 6	**identity {address { ipv4-address | ipv6-address} | fqdn name | email email-id | key-id key-id}** **Example:** `Router(config-ikev2-keyring-peer)# identity key-id 1234`	Identifies the IKEv2 peer through the following identities: • E-mail • FQDN • IPv4 address • Key ID The identity is available for key lookup on the IKEv2 responder only.
Step 7	**pre-shared-key {local| remote} {0| 6| line}** **Example:** `Router(config-ikev2-keyring-peer)# pre-shared-key key123`	Specifies the PSK for the peer. Enter the **local** or the **remote** keyword to specify an asymmetric PSK. By default, the PSK is symmetric.
Step 8	**exit** **Example:** `Router(config-ikev2-keyring-peer)# end`	Exits keyring peer configuration mode mode, and returns to configuration mode.

Example: Configure Tunnel Authentication Using PreShared Keys

This is a sample code for configuring tunnel authentication using preshared keys

```
crypto ikev2 keyring keys
 peer p1
 identity key-id 0388881001
 pre-shared-key cisco
 !
 peer p2
 identity key-id 0388882002
 pre-shared-key cisco
!
crypto ikev2 keyring HUB-KEY
peer SPOKES
address 0.0.0.0 0.0.0.0
pre-shared-key cisco
```

Configure the IKEv2 Profile for a Certificate

To configure the certificate for your IKEv2 profile, run the **crypto ikev2 profile IPROF** command. The following is a sample code for configuring the IKEv2 profile for a certificate.

```
Router(config)# crypto ikev2 profile IPROF-psk
 match identity remote any
 identity local key-id dhcp
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 nat force-encap
```

Configure an IPSec Profile

To configure an IPSec profile, run the **crypto ipsec profile IPROF** command. The following is a sample code for configuring an IPSec profile.

```
Router(config)# crypto ipsec profile IPROF
 set security-association idle-time 300
```

Enable VPN-SIP

To enable the VPN-SIP feature, run the **vpn-sip enable** command. The following is a sample code to enable VPN-SIP.

```
Router(config)# vpn-sip enable
vpn-sip local-number dhcp address ipv4 GigabitEthernet0/0/0
vpn-sip tunnel source Loopback1
```

Configure a LAN Side Interface

To configure a LAN side interface, run the **interface VLAN <interface>** command. The following is a sample code to configure a LAN side interface.

```
Router(config)# interface GigabitEthernet2
ip address 192.0.2.3 255.255.255.0
no shutdown
```

Configure a Loopback Interface

To configure a loopback interface, run the **interface loopback** <number> command. The following is a code sample to configure a loopback interface.

```
Router(config)# interface Loopback1
 ip address 10.255.255.3 255.255.255.0
 ip nat inside
```

Configure a Tunnel Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel source** {*ip-address* | *interface-type number*}
5. **tunnel destination**
6. **tunnel protection IPsec profile** *name*
7. **vpn-sip local-number dhcp remote-number bandwidth**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel1`	Configures a tunnel interface and enters the interface configuration mode. The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces that you can create.

Step 4	**tunnel source** {*ip-address* | *interface-type number*}
	Example:
	`Router(config-if)# ip address 12.12.12.12 255.255.255.255`
	`tunnel source Loopback1`
		Sets the source IP address or the source interface type number for a tunnel interface. Since the **tunnel protection IPsec profile** command is also used in this procedure, the tunnel source must specify an interface and not an IP address.

Step 5	**tunnel destination** **Example:** `Router(config-if)# tunnel destination destination dynamic`	Specifies the destination of the tunnel.

	Command or Action	Purpose
Step 6	**tunnel protection IPsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk`	Associates a tunnel interface with an IPsec profile. The *name* argument specifies the name of the IPsec profile. This value must match the name specified in the **crypto IPsec profile** <*name*>command.
Step 7	**vpn-sip local-number dhcp remote-number bandwidth** **Example:** `Router(config-if)# vpn-sip local-number dhcp remote-number 0388881001 bandwidth 1000`	Configures the interface for VPN-SIP. Bandwidth is the maximum data transmission rate that must be negotiated with this peer; the negotiated value is set on the tunnel interface. Choose one of these values—64, 128, 256, 512, or 1000 kbps. **Note** After you configure an interface for VPN-SIP, you cannot make any changes to the tunnel mode, tunnel destination, tunnel source, and tunnel protection. To change the mode, source, destination, or tunnel protection, you must remove the VPN-SIP configuration from the interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configure a Tunnel Interface

This is a sample code to configure a tunnel interface.

```
Router(config)# interface Tunnel1
 ip address 10.12.12.12 255.255.255.255
 tunnel source Loopback1
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk
 vpn-sip local-number dhcp remote-number 0388881001 bandwidth 1000
!
interface Tunnel10
 ip address 10.20.20.21 255.255.255.255
 tunnel source Loopback1
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk
 vpn-sip local-number dhcp remote-number 0388882002 bandwidth 100
```

Verify the DHCP Configuration in VPN-SIP

The following show command outputs indicate how to verify whether the DHCP in VPN-SIP is successfully configured on the Cisco IOS XE router behind the HGW.

```
Router_behind_HGW# show vpn-sip sip dhcp
SIP DHCP Info
SIP-DHCP interface:  GigabitEthernet 0/0/0
```

```
SIP server address:  ipv4:192.168.1.1
Domain name:         dns:ntt-east.ne.jp

Router_behind_HGW# show vpn-sip registration-status
  SIP registration of local number dhcp : registered 192.168.1.200
  Local dynamic number via dhcp[3], via SIP[0398765432]

Router_behind_HGW# show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================
3             ntt-east.ne.jp   2439          192.168.1.20
UDP           FFFFFFFFCCE6C415-5D8611ED-FFFFFFFF810AE9D4-FFFFFFFFD

Router_behind_HGW# show vpn-sip session detail
VPN-SIP session current status
Interface: Tunnel0
    Session status: SESSION_UP (I)
    Uptime        : 00:00:37
    Remote number : 0387654321
    Local number  : dhcp
    Remote address:port: aaa.bbb.ccc.ddd:27129
    Local address:port : 192.168.1.200:50026
    Crypto conn handle: 0x4000003D
    SIP Handle        : 0x4000001B
    SIP callID        : 301
    Configured/Negotiated bandwidth: 256/256 kbps
    Applied service policy:

Router_behind_HGW# show crypto session
Crypto session current status
Interface: Tunnel0
Profile: IPROF
Session status: UP-ACTIVE
Peer: aaa.bbb.ccc.ddd port 27129
   Session ID: 26
   IKEv2 SA: local 10.255.255.1/4500 remote aaa.bbb.ccc.ddd/27129 Active
   IPSEC FLOW: permit ip   0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
   Active SAs: 2, origin: crypto map

Router_behind_HGW# show crypto ikev2 sa
  IPv4 Crypto IKEv2  SA

Tunnel-id Local                 Remote                fvrf/ivrf
   Status
1         10.255.255.1/4500     aaa.bbb.ccc.ddd/27129  none/none
    READY
       Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH
       Grp:19, Auth sign: PSK, Auth verify: PSK
       Life/Active Time: 86400/86 sec
       CE id: 1022, Session-id: 22
       Local spi: 59E8EED28441BC32
       Remote spi: B5487716A19873BE
       IPv6 Crypto IKEv2  SA

Router_behind_HGW# show crypto ipsec sa

interface: Tunnel0
Crypto map tag: Tunnel0-head-0, local addr 10.255.255.1
protected vrf: (none)
local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
current_peer aaa.bbb.ccc.ddd port 27129
PERMIT, flags={origin_is_acl,}
     #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
     #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
     #pkts compressed: 0, #pkts decompressed: 0
```

```
     #pkts not compressed: 0, #pkts compr. failed: 0
     #pkts not decompressed: 0, #pkts decompress failed: 0
     #send errors 0, #recv errors 0

local crypto endpt.: 10.255.255.1, remote crypto endpt.:
aaa.bbb.ccc.ddd
plaintext mtu 1422, path mtu 1500, ip mtu 1500, ip mtu idb
GigabitEthernet0/0/0
current outbound spi: 0xE0F51D37(3774160183)
PFS (Y/N): N, DH group: none

inbound esp sas:
      spi: 0x493D896(76798102)
      transform: esp-aes esp-sha-hmac ,
      in use settings ={Tunnel UDP-Encaps, }
      conn id: 2044, flow_id: ESG:44, sibling_flags FFFFFFFF80004048,
crypto map: Tunnel0-head-0, initiator : True
       sa timing: remaining key lifetime (k/sec): (4607999/3509)
       IV size: 16 bytes
       replay detection support: Y
       Status: ACTIVE(ACTIVE)

      inbound ah sas:

      inbound pcp sas:

      outbound esp sas:
       spi: 0xE0F51D37(3774160183)
         transform: esp-aes esp-sha-hmac ,
         in use settings ={Tunnel UDP-Encaps, }
         conn id: 2043, flow_id: ESG:43, sibling_flags FFFFFFFF80004048,
crypto map: Tunnel0-head-0, initiator : True
         sa timing: remaining key lifetime (k/sec): (4607999/3509)
         IV size: 16 bytes
         replay detection support: Y
         Status: ACTIVE(ACTIVE)
      outbound ah sas:
      outbound pcp sas:


Router_behind_HGW# show ip nat translations
Pro  Inside global         Inside local          Outside local
Outside global
udp  192.168.1.200:50269   10.255.255.1:4500     aaa.bbb.ccc.ddd:23060
aaa.bbb.ccc.ddd:23060
Total number of translations: 1
```

Troubleshooting for VPN-SIP

Viewing Tunnel Interface in Show Output

Symptom

Show VPN-SIP session doesn't show any information about the tunnel interface. In the following example, information about the tunnel interface, tunnel1 is not shown:

```
Peer5-F#show vpn-sip session
VPN-SIP session current status

Interface: Tunnel2
   Session status: READY_TO_CONNECT
```

```
   Remote number : 0334563333
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel3
   Session status: READY_TO_CONNECT
   Remote number : 0323452222
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel4
   Session status: READY_TO_CONNECT
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel6
   Session status: READY_TO_CONNECT
   Remote number : 0634567777
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
```

Possible Cause

VPN-SIP is not configured on the tunnel interface

```
Peer5-F#sh run int tun1
Building configuration...

Current configuration : 201 bytes
!
interface Tunnel1
 ip address 10.5.5.5 255.0.0.0
 tunnel source Loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
end
```

Recommended Action

Configure VPN-SIP on the tunnel interface.

```
:

Peer5-F#show running interface tunnel 1
Building configuration...

Current configuration : 278 bytes
!
interface Tunnel1
 ip address 10.5.5.5 255.255.255.255
 tunnel source Loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
 vpn-sip local-number 0623458888 remote-number 0312341111 bandwidth 1000
end
```

Following is the running output for the above scenario:

```
Peer5-F#show vpn-sip session detail
VPN-SIP session current status

Interface: Tunnel1
   Session status: READY_TO_CONNECT
   Remote number : 0312341111
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0

   Crypto conn handle: 0x8000002C
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps

Interface: Tunnel2
   Session status: READY_TO_CONNECT
   Remote number : 0334563333
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000012
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 512/0 kbps

Interface: Tunnel3
   Session status: READY_TO_CONNECT
   Remote number : 0323452222
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000031
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 512/0 kbps

Interface: Tunnel4
   Session status: READY_TO_CONNECT
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x8000002F
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps

Interface: Tunnel6
   Session status: READY_TO_CONNECT
   Remote number : 0634567777
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000026
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps
```

Troubleshooting SIP Registration Status

Symptom

SIP registration status is Not Registered

```
Peer5#show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================

Peer5-F#show vpn-sip registration-status

 SIP registration of local number 0623458888 : not registered
```

Possible Cause

IP address is not configured on the WAN interface.

```
Peer5#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES unset  down                  down
GigabitEthernet0/1         unassigned      YES unset  up                    up
GigabitEthernet0/2         unassigned      YES unset  down                  down
GigabitEthernet0/3         unassigned      YES unset  down                  down
GigabitEthernet0/4         unassigned      YES unset  up                    up
GigabitEthernet0/5         10.5.5.5        YES manual up                    up
Vlan1                      10.45.1.5        YES NVRAM  up                     up
NVI0                       10.1.1.1         YES unset  up                     up
Loopback1                  10.1.1.1         YES NVRAM  up                     up
Loopback5                  10.5.5.5        YES NVRAM  administratively down down
Loopback11                 10.11.11.11     YES NVRAM  up                    up
Tunnel1                    10.5.5.5         YES NVRAM  up                     down
Tunnel2                    10.2.2.2         YES NVRAM  up                     down
Tunnel3                    10.3.3.3         YES NVRAM  up                     down
Tunnel4                    10.4.4.4         YES NVRAM  up                     down
Tunnel6                    10.8.8.8         YES NVRAM  up                     down

Peer5-F#show run interface gigabitEthernet 0/4
Building configuration...

Current configuration : 213 bytes
!
interface GigabitEthernet0/4
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 no ip address           ====> no IP address
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
end
```

Recommended Action

Use the **ip address dhcp** command to configure the interface IP address.

```
Peer5-F#show running-config interface gigabitEthernet 0/4
Building configuration...

Current configuration : 215 bytes
!
interface GigabitEthernet0/4
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp         ====> configure IP address DHCP
 ip nat outside
 ip virtual-reassembly in
 duplex auto
```

```
 speed auto
end


Peer5-F#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES unset  down                  down
GigabitEthernet0/1         unassigned      YES unset  up                    up
GigabitEthernet0/2         unassigned      YES unset  down                  down
GigabitEthernet0/3         unassigned      YES unset  down                  down
GigabitEthernet0/4         172.30.18.22    YES DHCP   up                    up
GigabitEthernet0/5         10.5.5.5        YES manual up                    up
Vlan1                      10.45.1.5        YES NVRAM  up                    up
NVI0                       10.1.1.1         YES unset  up                    up
Loopback1                  10.1.1.1         YES NVRAM  up                    up
Loopback5                  10.5.5.5        YES NVRAM  administratively down down
Loopback11                 10.11.11.11     YES NVRAM  up                    up
Tunnel1                    10.6.5.5         YES NVRAM  up                    down
Tunnel2                    10.2.2.2         YES NVRAM  up                    down
Tunnel3                    10.3.3.3         YES NVRAM  up                    down
Tunnel4                    10.4.4.4         YES NVRAM  up                    down
Tunnel6                    10.8.8.8         YES NVRAM  up                    down

Peer5-F#show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================
0623458888    example.com   2863          172.30.18.22
UDP           1E83ECF0-AF0611E7-802B8FCF-594EB9E7@122.50.18.22

Peer5-F#show vpn-sip registration-status

 SIP registration of local number 0623458888 : registered 172.30.18.22
```

Session stuck in Negotiating IKE state

Symptom

VPN-SIP session stuck in Negotiating IKE state.

```
Peer5#show vpn-sip session remote-number 0612349999 detail
VPN-SIP session current status

Interface: Tunnel4
   Session status: NEGOTIATING_IKE (R)
   Uptime        : 00:00:58
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 172.30.168.3:24825
   Local address:port : 172.30.18.22:50012
   Crypto conn handle: 0x8000002E
   SIP Handle        : 0x8000000C
   SIP callID        : 16
   Configured/Negotiated bandwidth: 1000/1000 kbps
```

Possible Cause

Bad configuration related to IKEv2.

In the following example the Key ID that is configured in the keyring does not match the SIP numberof the remote peer.

```
Peer5-F#show running-config interface tunnel 4
Building configuration...
```

```
Current configuration : 276 bytes
!
interface Tunnel4
 ip address 10.4.4.4 255.0.0.0
 tunnel source Loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
 VPN-SIP local-number 0623458888 remote-number 0612349999 bandwidth 1000  ====> Remote
number mentioned here doesn’t match the remote number in the keyring
end

IKEv2 Keyring configs:
!
crypto ikev2 keyring keys
 peer peer1
  identity key-id 0312341111
  pre-shared-key psk1
 !
 peer abc
  identity key-id 0345674444
  pre-shared-key psk1
 !
 peer peer2
  identity key-id 0334563333
  pre-shared-key psk10337101690
 !
 peer peer6
  identity key-id 0634567777
  pre-shared-key cisco123
 !
 peer peer3
  identity key-id 0323452222
  pre-shared-key cisco123
 !
 peer peer4
  identity key-id 0645676666
  pre-shared-key psk1
 !
 peer NONID
  identity fqdn example.com
  pre-shared-key psk1
 !
!
!
crypto ikev2 profile test
 match identity remote any
 identity local key-id 0623458888
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 dpd 10 6 periodic
 nat force-encap
```

Recommended Action

Correct the keyring configurations.

```
rypto ikev2 keyring keys
 peer peer1
  identity key-id 0312341111
  pre-shared-key psk1
 !
 peer abc
```

```
  identity key-id 0345674444
  pre-shared-key psk1
 !
 peer peer2
  identity key-id 0334563333
  pre-shared-key psk1
 !
 peer peer6
  identity key-id 0634567777
  pre-shared-key psk1
 !
 peer peer3
  identity key-id 0323452222
  pre-shared-key psk1
 !
 peer peer4
  identity key-id 0612349999
  pre-shared-key psk1
 !
 peer NONID
  identity fqdn example.com
  pre-shared-key psk1
 !
!
!
crypto ikev2 profile test
 match identity remote any
 identity local key-id 0623458888
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 dpd 10 6 periodic
 nat force-encap
!

Peer5-F#show vpn-sip session remote-number  0612349999 detail
VPN-SIP session current status

Interface: Tunnel4
   Session status: SESSION_UP (R)
   Uptime        : 00:02:04
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 172.30.168.3:24845
   Local address:port : 172.30.18.22:50020
   Crypto conn handle: 0x8000004E
   SIP Handle        : 0x80000014
   SIP callID        : 24
   Configured/Negotiated bandwidth: 1000/1000 kbps
```

Troubleshooting Session Initiation

Symptom

Session does not initiate and gets stuck in Negotiating IKE state

Possible Cause

Fagmentation of IKE packets when a large PKI certificate is included in the IKE authentication message.

Recommended Action

Configure IKEv2 fragmentation on the routers.

Debug Commands

The folllwing debug commands are available to debug VPN-SIP configuration:

Table 213: debug commands

Command Name	Description
debug vpn-sip event	Prints debug messages for SVTI registration with VPN-SIP, SIP registration, call setup, and so on.
debug vpn-sip errors	Prints error messages only when an error occurs during initialization, registration, call setup, and so on.
debug vpn-sip sip all	Enables all SIP debugging traces.
debug vpn-sip sip calls	Enables SIP SPI calls debugging trace.
debug vpn-sip sip dhcp	Enables SIP-DHCP debugging trace
debug vpn-sip sip error	Enables SIP error debugging trace
debug vpn-sip sip events	Enables SIP events debugging trace.
debug vpn-sip sip feature	Enables feature level debugging.
debug vpn-sip sip function	Enables SIP function debugging trace.
debug vpn-sip sip info	Enables SIP information debugging trace.
debug vpn-sip sip level	Enables information level debugging.
debug vpn-sip sip media	Enables SIP media debugging trace.
debug vpn-sip sip messages	Enables SIP SPI messages debugging trace
debug vpn-sip sip non-call	Enables Non-Call-Context trace (OPTIONS, SUBSCRIBE, and so on)
debug vpn-sip sip preauth	Enable SIP preauth debugging trace.
debug vpn-sip sip states	Enable SIP SPI states debugging trace.
debug vpn-sip sip translate	Enables SIP translation debugging trace.
debug vpn-sip sip transport	Enables SIP transport debugging traces.
debug vpn-sip sip verbose	Enables verbose mode.

Additional References for VPN-SIP

Standards and RFCs

Standard/RFC	Title
RFC 6193 (with Restrictions)	Media Description for the Internet Key Exchange Protocol (IKE) in the Session Description Protocol (SDP)

CHAPTER 159

Deleting Crypto Sessions of Revoked Peer Certificates

The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL.

Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates

- If revocation check is turned off and this feature is enabled, the IKE database is not populated with the number of sessions. The show outputs do not display information about the deleted sessions.
- Frequent enabling and disabling of this feature (with active sessions on the device) is not recommended.
- Frequent CRL downloads (in a span of 30 minutes) for the same issuername (CA server) is not recommended.
- CRL cache must be enabled. CRL caching cannot be disabled for trustpoint-based prefetch. However, it is possible to disable CRL caching for URL-based prefetch.
- In case of autoenrollment on IKE, the sessions are not deleted until the next IKE rekey, whereas in case of IKEv2, the tunnel must be cleared manually or wait until the certificate expires.
- If IKE has database of “issuer-name” and “SN” populated and receives a notification from PKI about certificate revocation, IKE would act on the PKI notification.

Information About Deleting Crypto Sessions of Revoked Peer Certificates

How a Crypto Session is Deleted

1. When negotiating via certificate authentication, the peer sends the CERT payload to the device, whcih parses each certificate to store information about serial number and the issuer names. This information forms the list of serial numbers issued by the corresponding CA server and is passed to PKI for revocation check.
2. If the revocation-check crl command is configured for a trustpoint, PKI informs IKE about the revocation check thereby disabling IKE from unnecessarily storing unwanted peer certification information.
3. After a successful CRL download, PKI sends IKE a notification, which contains the “issuer-name.” The CRL signature and content is verified. If there is no change in CRL content, PKI does not notify IKE.
4. If PKI notifies IKE containing the issuer name, IKE prepares a list of serial numbers for an issuer name and passes this list to PKI to verify if the serial numbers in the list are revoked.
5. PKI performs revocation check on the serial number list received from the IKE and checks the list against the downloaded CRL. The revoked serial number list is returned to IKE.
6. On a notification from PKI containing the list of revoked serial numbers, IKE identifies and deletes sessions pertaining to those serial numbers those sessions.

How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates

Enabling Deletion of Crypto Sessions

Perform this task to enable the deletion of crypto sessions for revoked certificates.

SUMMARY STEPS

1. **enable**
2. **clear crypto session**
3. **configure terminal**
4. Do one of the following:
   - **crypto isakmp disconnect-revoked-peers**
   - **crypto ikev2 disconnect-revoked-peers**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** **Example:** `Device# clear crypto session`	(Optional) Deletes IPsec crypto sessions and IKE and security associations. **Note** Use this command to enable the feature for previously established sessions, else the feature is enabled for new sessions only.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 4	Do one of the following:
	• **crypto isakmp disconnect-revoked-peers**
	• **crypto ikev2 disconnect-revoked-peers**
	Example:
	`Device(config)# crypto isakmp disconnect-revoked-peers`
	Example:
	`Device(config)# crypto ikev2 disconnect-revoked-peers`
		Disconnects IKE or IKEv2 crypto sessions with peers having revoked certificates.
		For this command to take effect, reconnected the existing sessions.

Step 5	**end** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the Delete Crypto Session Capability for a Revoked Peer Certificate

Perform this task to verify if the delete crypto session capability is displayed in the show output.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp peers**
3. **show crypto ikev2 session detail**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show crypto isakmp peers**

Example:

```
Device# show crypto isakmp peers
```

Displays Internet Security Association and Key Management Protocol (ISAKMP) peer descriptions.

Step 3 **show crypto ikev2 session detail**

Example:

```
Device# show crypto ikev2 session detail
```

Displays the status of active Internet Key Exchange Version 2 (IKEv2) sessions.

Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates

Example: Enabling Deletion of Crypto Sessions for an IKE Session

```
Device> enable
Device# clear crypto session
Device# configure terminal
Device(config)# crypto isakmp disconnect-revoked-peers
Device# show crypto isakmp peers

Peer: 150.1.1.2 Port: 500 Local: 150.1.1.1
 Phase1 id: 150.1.1.2
 Disconnect Revoked Peer: Enabled
```

Example: Enabling Deletion of Crypto Sessions for an IKEv2 Session

```
Device> enable
Device# clear crypto session
Device# configure terminal
Device(config)# crypto ikev2 disconnect-revoked-peers
Device# show crypto ikev2 session detail

Session-id:1, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id   Local          Remote           fvrf/ivrf          Status
1           10.0.0.1/500    10.0.0.2/500      (none)/(none)      READY
      Encr: 3DES, Hash: SHA96, DH Grp:2, Auth: PSK
      Life/Remaining/Active Time: 86400/86157/248 sec
      CE id: 0, Session-id: 1, MIB-id: 1
      Status Description: Negotiation done
```

```
      Local spi: 750CBE827434A245       Remote spi: 4353FEDBABEBF24C
      Local id:          10.0.0.1        Remote id:          10.0.0.2
      Local req mess id:    0           Remote req mess id: 0
      Local next mess id:    0          Remote next mess id: 2
      Local req queued:    0            Remote req queued: 0
      Local window:    5                 Remote window: 5
      DPD configured for 0 seconds
      NAT-T is not detected
      Disconnect Revoked Peer: Enabled
Child sa: local selector  10.0.0.1/0 - 10.0.0.1/65535
          remote selector 10.0.0.2/0 - 10.0.0.2/65535
          ESP spi in/out: 0x9360A95/0x6C340600
          CPI in/out: 0x9FE5/0xC776
          AH spi in/out: 0x0/0x0
          Encr: AES CBC, keysize: 128, esp_hmac: SHA96
          ah_hmac: Unknown - 0, comp: IPCOMP_LZS, mode tunnel
```

Additional References for Deleting Crypto Sessions of Revoked Peers

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Configuring IKE	*Configuring Internet Key Exchange for IPsec VPNs*
Configuring IKEv2	*Configuring Internet Key Exchange Version 2 and FlexVPN Site-to-Site*
Recommended cryptographic algorithms	*Next Generation Encryption*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 214: Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates

Feature Name	Releases	Feature Information
Delete crypto session(s) of revoked peer cert(s) on CRL download		The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL. The following commands were introduced or modified: **crypto ikev2 disconnect-revoked-peers**, **crypto isakmp disconnect-revoked-peers**, **show crypto isakmp peers**, **show crypto ikev2 session detail**.

CHAPTER 160

Crypto Conditional Debug Support

The Crypto Conditional Debug Support feature introduces new debug commands that allow users to debug an IP Security (IPsec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPsec operations and reducing the amount of debug output, users can better troubleshoot a router with a large number of tunnels.

Prerequisites for Crypto Conditional Debug Support

Restrictions for Crypto Conditional Debug Support

-
- Although conditional debugging is useful for troubleshooting peer-specific or functionality related Internet Key Exchange (IKE) and IPSec problems, conditional debugging may not be able to define and check large numbers of debug conditions. Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a router with heavy traffic should be used with caution.

Information About Crypto Conditional Debug Support

Supported Condition Types

The new crypto conditional debug CLIs--**debug crypto condition**, **debug crypto condition unmatched**, **and show crypto debug-condition**--allow you to specify conditions (filter values) in which to generate and

display debug messages related only to the specified conditions. The table below lists the supported condition types.

Note The **debug crypto condition peer** command with the **ipv4** or **ipv6** keyword can provide the hardware platform specific debugging output. The rest of the condition filters do not provide platform specific debugging output.

Table 215: Supported Condition Types for Crypto Debug CLI

Condition Type (Keyword)	Description
connid [21]	An integer between 1-32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the connection ID to interface with the crypto engine.
FVRF	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its front-door VRF (FVRF).
ikev2	The name string for an IKEv2 profile. Relevant debug messages will be shown if the IKEv2 profile name is specified.
isakmp	The name string for an ISAKMP profile. Relevant debug messages will be shown if the ISAKMP profile name is specified.
IVRF	The name string of a VRF instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its inside VRF (IVRF).
local	The name string of an IPv4 or IPv6 local address.
peer group	A Unity group-name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity; for example, if the peer is enabling IKE Xauth with this FQDN string.
peer ipv4 or peer ipv6	A single IP address. Relevant debug messages will be shown if the current IPsec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPsec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity; for example, if the peer is enabling IKE Extended Authentication (Xauth) with this username.
session	Provides information about crypto sessions.
SPI	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPSec operation uses this value as the SPI.

Condition Type (Keyword)	Description
unmatched	Provides debug messages when context information is unavailable.

[21] If an IPSec connid, flowid, or SPI is used as a debug condition, the debug messages for a related IPSec flow are generated. An IPSec flow has two connids, flowids, and SPIs--one inbound and one outbound. Both two connids, flowids, and SPIs can be used as the debug condition that triggers debug messages for the IPSec flow.

How to Enable Crypto Conditional Debug Support

Enabling Crypto Conditional Debug Messages

Performance Considerations

- Before enabling crypto conditional debugging, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.

Note Specifying numerous debug conditions may consume CPU cycles and negatively affect router performance.

- Your router will perform conditional debugging only after at least one of the global crypto debug commands--**debug crypto isakmp**, **debug crypto ipsec**, and **debug crypto engine**--has been enabled. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Disable Crypto Debug Conditions

If you choose to disable crypto conditional debugging, you must first disable any crypto global debug CLIs you have issued ; thereafter, you can disable conditional debugging.

Note The **reset** keyword can be used to disable all configured conditions at one time.

SUMMARY STEPS

1. **enable**
2. **debug crypto condition** [**connid** *integer* **engine-id** *integer*] [**flowid** *integer***engine-id** *integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**]
3. **show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]}
4. **debug crypto isakmp**
5. **debug crypto ipsec**

6. **debug crypto engine**
7. **debug crypto condition unmatched** [**isakmp** | **ipsec** | **engine**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto condition** [**connid** *integer* **engine-id** *integer*] [**flowid** *integer***engine-id** *integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**] **Example:** `Router# debug crypto condition connid 2000 engine-id 1`	Defines conditional debug filters.
Step 3	**show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]} **Example:** `Router# show crypto debug-condition spi`	Displays crypto debug conditions that have already been enabled in the router.

Step 4	**debug crypto isakmp**
	Example:
	`Router#`
	`debug crypto isakmp`
		Enables global IKE debugging.

Step 5	**debug crypto ipsec**
	Example:
	`Router#`
	`debug crypto ipsec`
		Enables global IPSec debugging.

Step 6	**debug crypto engine**
	Example:
	`Router#`
	`debug crypto engine`
		Enables global crypto engine debugging.

Step 7	**debug crypto condition unmatched** [**isakmp**	**ipsec**	**engine**] **Example:**	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions.

	Command or Action	Purpose
	`Router# debug crypto condition unmatched ipsec`	If none of the optional keywords are specified, all crypto-related information will be shown.

Enabling Crypto Error Debug Messages

To enable crypto error debug messages, you must perform the following tasks.

debug crypto error CLI

Enabling the **debug crypto error** command displays only error-related debug messages, thereby, allowing you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system.

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp** | **ipsec** | **engine**} **error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** | **ipsec** | **engine**} **error** **Example:** `Router# debug crypto ipsec error`	Enables only error debugging messages for a crypto area.

Configuration Examples for the Crypto Conditional Debug CLIs

Enabling Crypto Conditional Debugging Example

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3, and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to enable global debug crypto CLIs and enable the **show crypto debug-condition** command to verify conditional settings.

```
Router#
debug crypto condition connid 2000 engine-id 1
Router#
debug crypto condition peer ipv4 10.1.1.1
Router#
debug crypto condition peer ipv4 10.1.1.2
Router#
debug crypto condition peer ipv4 10.1.1.3
Router#
debug crypto condition unmatched
! Verify crypto conditional settings.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON
IKE peer IP address filters:
10.1.1.1  10.1.1.2   10.1.1.3
Connection-id filters:[connid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router#
debug crypto isakmp
Router#
debug crypto ipsec
Router#
debug crypto engine
```

Disabling Crypto Conditional Debugging Example

The following example shows how to disable all crypto conditional settings and verify that those settings have been disabled:

```
Router#
debug crypto condition reset
! Verify that all crypto conditional settings have been disabled.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

Additional References

The following sections provide references to the Crypto Conditional Debug Support feature.

Related Documents

Related Topic	Document Title
IPSec and IKE configuration tasks	" Internet Key Exchange for IPsec VPNs " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
IPSec and IKE commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Crypto Conditional Debug Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 161

IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature allows both IPv6 unicast and multicast traffic to pass through a protected generic routing encapsulation (GRE) tunnel.

Prerequisites for IPv6 over IPv4 GRE Tunnel Protection

- To enable this feature, you must configure IPsec tunnel protection on an IPv4 GRE tunnel.
- To enable IPv6 multicast, you must configure IPv6 multicast routing.

Restrictions for IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature supports IPv6 over IPv4 point-to-point GRE tunnel protection and not IPv6 over IPv4 mGRE tunnel protection.

Information About IPv6 over IPv4 GRE Tunnel Protection

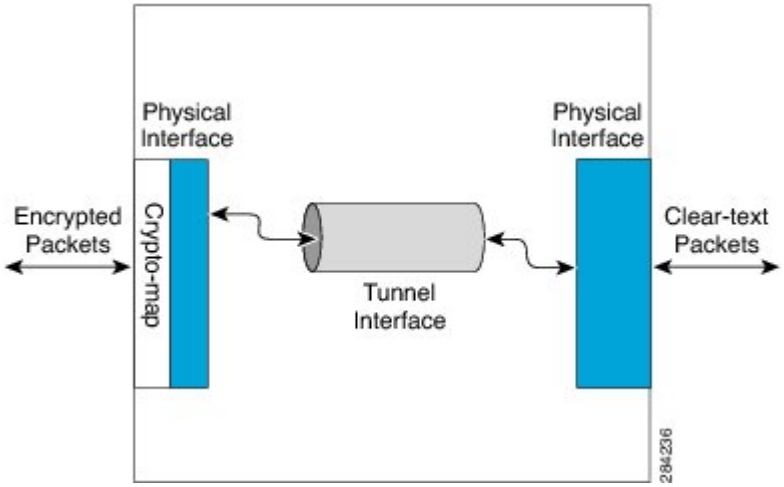
GRE Tunnels with IPsec

Generic routing encapsulation (GRE) tunnels sometimes are combined with IPSec, because IPSec does not support IPv6 multicast packets. This function prevents dynamic routing protocols from running successfully over an IPSec VPN network. Because GRE tunnels do support IPv6 multicast , a dynamic routing protocol can be run over a GRE tunnel. Once a dynamic routing protocol is configured over a GRE tunnel, you can encrypt the GRE IPv6 multicast packets using IPSec.

IPSec can encrypt GRE packets using a crypto map or tunnel protection. Both methods specify that IPSec encryption is performed after GRE encapsulation is configured. When a crypto map is used, encryption is applied to the outbound physical interfaces for the GRE tunnel packets. When tunnel protection is used, encryption is configured on the GRE tunnel interface.

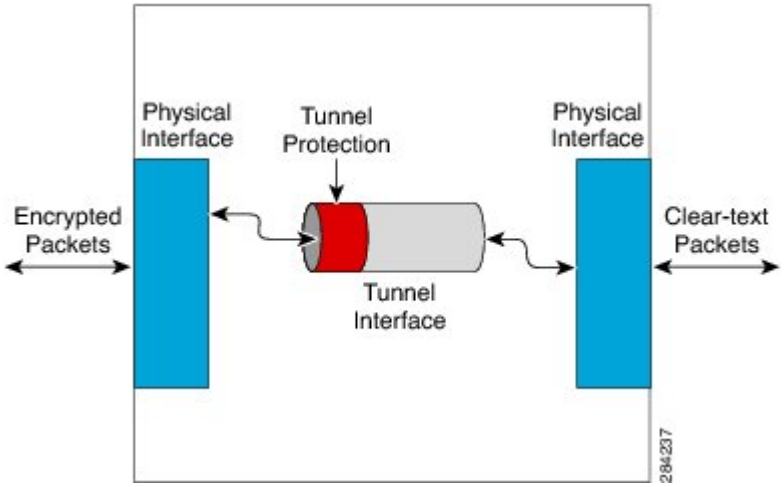
The following figure shows encrypted packets that enter a router through a GRE tunnel interface using a crypto map on the physical interface. Once the packets are decrypted and decapsulated, they continue to their IP destination as clear text.

Figure 83: Using a Crypto Map to Configure IPv6 over IPv4 GRE Tunnel Encryption



The following figure shows encryption using tunnel protection command on the GRE tunnel interface. The encrypted packets enter the router through the tunnel interface and are decrypted and decapsulated before they continue to their destination as clear text.

Figure 84: Using Tunnel Protection to Configure IPv6 over IPv4 GRE Tunnel Encryption



There are two key differences in using the crypto map and tunnel protection methods:

- The IPSec crypto map is tied to the physical interface and is checked as packets are forwarded out through the physical interface. At this point, the GRE tunnel has already encapsulated the packet.

- Tunnel protection ties the encryption functionality to the GRE tunnel and is checked after the packet is GRE encapsulated but before the packet is handed to the physical interface.

How to Configure IPv6 over IPv4 GRE Tunnel Protection

Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing**
4. **ipv6 unicast-routing**
5. **interface** *type number*
6. **ipv6 address** {**ipv6-address**/*prefix-length* | **prefix-name** *sub-bits*/*prefix-length*}
7. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ip** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}
8. **tunnel source** {**ip-address** | **ipv6-address** | *interface-typeinterface-number*}
9. **tunnel destination** {*hostname* | *ip-address* | *ipv6-address*}
10. **exit**
11. **crypto isakmp policy** *priority*
12. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
13. **hash** {**sha** | **md5**}
14. **group** {**1** | **2** | **5**}
15. **encryption** {**des** | **3des** | **aes 192** | **aes 256**}
16. **exit**
17. **crypto isakmp key** *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address*/*ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
18. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
19. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]]
20. **crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp** [**dynamic** *dynamic-map-name* | **discover** | **profile** *profile-name*]]
21. **set peer** {*hostname* [**dynamic**] [**default**] | *ip-address* [**default**]}
22. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
23. **match address** [*access-list-id* | *name*]
24. **exit**
25. **interface** *type number*
26. **crypto map** *map-name* [**redundancy** *standby-group-name* [**stateful**]]
27. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** **Example:** `Router(config)# ipv6 multicast-routing`	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces of the router and enables multicast forwarding. • Enable this command only if you are using IPv6 multicast. If you are using IPv6 unicast, you need not enable this command.
Step 4	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 5	**interface** *type number* **Example:** `Router(config)# interface tunnel 10`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 6	**ipv6 address** {**ipv6-address**/*prefix-length* | **prefix-name** *sub-bits*/*prefix-length*} **Example:** `Router(config-if)# ipv6 address 0:0:0:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ip** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Example:** `Router(config-if)# tunnel mode gre ip`	Sets the encapsulation mode for the tunnel interface.
Step 8	**tunnel source** {**ip-address** | **ipv6-address** | *interface-typeinterface-number*} **Example:** `Router(config-if)# tunnel source ethernet0`	Sets the source address for a tunnel interface.
Step 9	**tunnel destination** {*hostname* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 172.16.0.12`	Specifies the destination for a tunnel interface.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 15`	Defines an Internet Key Exchange (IKE) policy, and enters ISAKMP policy configuration mode. • Policy number 1 indicates the policy with the highest priority. The lower the *priority* argument value, the higher the priority.
Step 12	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Example:** `Router(config-isakmp-policy)# authentication pre-share`	Specifies the authentication method within an IKE policy. • The **rsa-sig** and **rsa-encr** keywords are not supported in IPv6.
Step 13	**hash** {**sha** | **md5**} **Example:** `Router(config-isakmp-policy)# hash md5`	Specifies the hash algorithm within an IKE policy.
Step 14	**group** {**1** | **2** | **5**} **Example:** `Router(config-isakmp-policy)# group 2`	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 15	**encryption** {**des** | **3des** | **aes 192** | **aes 256**} **Example:** `Router(config-isakmp-policy)# encryption 3des`	Specifies the encryption algorithm within an IKE policy.
Step 16	**exit** **Example:** `Router(config-isakmp-policy)# exit`	Exits ISAKMP policy configuration mode and enters global configuration mode.
Step 17	**crypto isakmp key** *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**] **Example:** `Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0`	Configures a preshared authentication key.
Step 18	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des`	Defines a transform set.
Step 19	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source*	Defines an extended IP access list.

	Command or Action	Purpose
	source-wildcard destination destination-wildcard [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]] **Example:** `Router(config)# access-list 110 permit gre host 192.168.0.16 host 172.16.0.12`	
Step 20	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp** [**dynamic** *dynamic-map-name* | **discover** | **profile** *profile-name*]] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Creates a new crypto map entry or profile and enters crypto map configuration mode.
Step 21	**set peer** {*hostname* [**dynamic**] [**default**] | *ip-address* [**default**]} **Example:** `Router(config-crypto-map)# set peer 10.0.0.1`	Specifies an IP Security (IPsec) peer in a crypto map entry.
Step 22	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-map)# set transform-set myset0`	Specifies the transform set that can be used with the crypto map entry.
Step 23	**match address** [*access-list-id* | *name*] **Example:** `Router(config-crypto-map)# match address 102`	Specifies an extended access list for a crypto map entry.
Step 24	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 25	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and number and enters interface configuration mode.
Step 26	**crypto map** *map-name* [**redundancy** *standby-group-name* [**stateful**]] **Example:** `Router(config-if)# crypto map mymap`	Applies a previously defined crypto map set to an outbound interface.
Step 27	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing**
4. **ipv6 unicast-routing**
5. **crypto isakmp policy** *priority*
6. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
7. **hash** {**sha** | **md5**}
8. **group** {**1** | **2** | **5**}
9. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
10. **exit**
11. **crypto isakmp key** *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
12. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
13. **crypto ipsec profile** *profile-name*
14. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
15. **exit**
16. **interface** *type number*
17. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length*}
18. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ip** | **gre ipv6** | **ipip**[**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}
19. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
20. **tunnel destination** {*hostname* | *ip-address* | *ipv6-address*}
21. **tunnel protection ipsec profile** *name* [**shared**]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** **Example:** `Router(config)# ipv6 multicast-routing`	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces of the router and enables multicast forwarding.

	Command or Action	Purpose
		• Enable this command only if you are using IPv6 multicast. If you are using IPv6 unicast, you do not need to enable this command.
Step 4	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 5	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 15`	Defines an IKE policy, and enters ISAKMP policy configuration mode. Policy number 1 indicates the policy with the highest priority. The lower the *priority* argument value, the higher the priority.
Step 6	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Example:** `Router(config-isakmp-policy)# authentication pre-share`	Specifies the authentication method within an Internet Key Exchange (IKE) policy. • The **rsa-sig** and **rsa-encr** keywords are not supported in IPv6.
Step 7	**hash** {**sha** | **md5**} **Example:** `Router(config-isakmp-policy)# hash md5`	Specifies the hash algorithm within an IKE policy.
Step 8	**group** {**1** | **2** | **5**} **Example:** `Router(config-isakmp-policy)# group 2`	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 9	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**} **Example:** `Router(config-isakmp-policy)# encryption 3des`	Specifies the encryption algorithm within an IKE policy.
Step 10	**exit** **Example:** `Router(config-isakmp-policy)# exit`	Exits ISAKMP policy configuration mode and returns to global configuration mode.
Step 11	**crypto isakmp key** *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address*/*ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**] **Example:** `Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0`	Configures a preshared authentication key.
Step 12	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:**	Defines a transform set, and places the router in crypto transform configuration mode.

	Command or Action	Purpose
	`Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des`	
Step 13	**crypto ipsec profile** *profile-name* **Example:** `Router(config)# crypto ipsec profile ipsecprof`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers and enters IPsec profile configuration mode.
Step 14	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(ipsec-profile)# set transform-set myset0`	Specifies the transform set that can be used with the crypto map entry.
Step 15	**exit** **Example:** `Router(ipsec-profile)# exit`	Exits IPsec profile configuration mode and returns to global configuration mode.
Step 16	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 17	**ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 18	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ip** | **gre ipv6** | **ipip**[**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Example:** `Router(config-if)# tunnel mode gre ip`	Specifies a GRE IPv6 tunnel.
Step 19	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source 10.0.0.1`	Specifies the source address or the source interface type and number for the tunnel interface.
Step 20	**tunnel destination** {*hostname* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 172.16.0.12`	Specifies the destination address or hostname for the tunnel interface.
Step 21	**tunnel protection ipsec profile** *name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile ipsecprof`	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto IPsec profile** *name* command.

	Command or Action	Purpose
		• The **shared** keyword allows IPsec sessions to be shared between multiple tunnel interfaces configured with the same tunnel source IP. **Note** When you modify the tunnel protection for an IPsec profile, you must shut down the tunnel interface first. After the modification is successful, you must manually turn on the tunnel configuration.
Step 22	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection

Example: Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

```
Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config)# interface tunnel 10
Router(config-if)# ipv6 address my-prefix 0:0:0:7272::72/64
Router(config-if)# tunnel mode gre ip
Router(config-if)# tunnel source ethernet0
Router(config-if)# tunnel destination 172.16.0.12
Router(config-if)# exit
Router(config)# crypto isakmp policy 15
Router(config-isakmp-policy)# authentication pre-share
Router(config-isakmp-policy)# hash md5
Router(config-isakmp-policy)# group 2
Router(config-isakmp-policy)# encryption 3des
Router(config-isakmp-policy)# exit
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0
Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des
Router(config)# access-list 110 permit gre host 192.168.0.16 host 172.16.0.12
Router(config)# crypto map mymap 10 ipsec-isakmp
Router(config-crypto-map)# set peer 10.0.0.1
Router(config-crypto-map)# set transform-set myset0
Router(config-crypto-map)# match address 102
Router(config-crypto-map)# exit
Router(config)# interface ethernet1
Router(config-if)# crypto map mymap
Router(config-if)# end
```

Example: Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

The following example configures IPsec tunnel protection on an IPv4 GRE tunnel. IPv6 multicast routing is enabled using the **ipv6 multicast-routing** command.

```
Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config)# crypto isakmp policy 15
Router(config-isakmp-policy)# authentication pre-share
Router(config-isakmp-policy)# hash md5
Router(config-isakmp-policy)# group 2
Router(config-isakmp-policy)# encryption 3des
Router(config-isakmp-policy)# exit
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0
Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des
Router(config)# crypto ipsec profile ipsecprof
Router(ipsec-profile)# set transform-set myset0
Router(ipsec-profile)# exit
Router(config)# interface tunnel 1
Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127
Router(config-if)# tunnel mode gre ip
Router(config-if)# tunnel source 10.0.0.1
Router(config-if)# tunnel destination 172.16.0.12
Router(config-if)# tunnel protection ipsec profile ipsecprof
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 Multicast Routing	IPv6 Implementation Guide
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 over IPv4 GRE Tunnel Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 162

RFC 430x IPsec Support

The RFC 430x IPsec Support includes features—RFC 430x IPsec Support Phase 1 and RFC430x IPsec Support Phase 2—that implement Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

Information About RFC 430x IPsec Support

RFC 430x IPsec Support Phase 1

The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

RFC 4301 specifies the base architecture for IPsec-compliant systems. RFC 4301 describes how to provide a set of security services for traffic at the IP layer, in both the IPv4 and IPv6 environments. The RFC 430x IPsec Support Phase 1 feature provides support for the following RFC 4301 implementations on Cisco IOS software.

- **Security association (SA) lifetime**—The lifetime of a security association between IPsec and Internet Key Exchange (IKE) or Internet Key Exchange Version 2 (IKEv2) must not exceed the lifetime of the authentication certificate.
- **OPAQUE selectors**—OPAQUE indicates that the corresponding selector field is not available for verification. When IKEv2 encounters an OPAQUE selector, IKEv2 skips, does not process the OPAQUE selector, and moves to next selector for policy verification.
- **Explicit Congestion Notification (ECN) support**—ECN is propagated when decrypting an IPsec packet thereby ensuring the packet source and destination are aware of congestion that occurs within the network.
- **Fragment processing**—Peers must not send Initial and noninitial fragments in the same tunnel. There must be a separate tunnel mode SA for carrying initial and noninitial fragments and separate tunnel mode SA for noninitial fragments. IPsec peers must support discarding of packets and stateful fragment checking to accommodate bypass traffic.
- **Do not fragment-(DF) bit processing**—DF-bit processing must be set on a per SA basis.

- **Dummy packet generation support**—It should be possible to send dummy packets via IPsec SA to encapsulate the packets when traffic is flowing via IPsec SA tunnel.

RFC 430x IPsec Support Phase 2

The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.

ICMP error messages are sent when an ICMP error occurs. For example, when a host is not reachable, the intermediate device sends a message to the originator of the ICMP request that the host is not reachable. When an ICMP error message reaches an IPsec encryption policy, it may not be classified to match an existing SA. So, the packets are classified based on the data inside the ICMP error message. This data contains the source and destination address of the original ICMP message. If an SA is found based on the address in the ICMP error message, the SA is used. If there is no SA, an SA is created if the policy permits. For decryption, the post decrypt check is performed on the data inside the ICMP error message if a valid SA is not found.

The encryption and decryption of ICMP error messages can be verified through the encrypt and decrypt counters displayed in the output of the **show crypto ipsec sa** command.

How to Configure RFC 430x IPsec Support

Configuring RFC 430x IPsec Support Globally

Perform this task to configure the RFC 4301 implementations globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*}
4. **crypto ipsec security-association ecn** {**discard** | **propogate**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:**	Enables the generation and transmission of dummy packets in an IPsec traffic flow.

	Command or Action	Purpose
	`Device(config)# crypto ipsec security-association dummy seconds 5`	
Step 4	**crypto ipsec security-association ecn** {**discard** | **propogate**} **Example:** `Device(config)# crypto ipsec security-association ecn discard`	Enables the Explicit Congestion Notification (ECN) settings in an IPsec traffic flow.
Step 5	**exit** **Example:** `Device(config-crypto-map)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring RFC 430x IPsec Support Per Crypto Map

Perform this task to configure the RFC 4301 implementations per crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set ipsec security-association dfbit** {**clear** | **copy** | **set**}
5. **set ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*}
6. **set ipsec security-association ecn** {**discard** | **propogate**}
7. **end**
8. **show crypto map ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Device(config)# crypto map cmap 1 ipsec-isakmp`	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.
Step 4	**set ipsec security-association dfbit** {**clear** | **copy** | **set**} **Example:**	Enables do not fragment (DF)-bit processing per security association (SA) for an IPsec traffic flow in a crypto map.

	Command or Action	Purpose
	`Device(config-crypto-map)# set ipsec security-association dfbit set`	
Step 5	**set ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:** `Device(config-crypto-map)# set ipsec security-association dummy seconds 5`	Enables the generation and transmission of dummy packets for an IPsec traffic flow in a crypto map.
Step 6	**set ipsec security-association ecn** {**discard** | **propogate**} **Example:** `Device(config-crypto-map)# set ipsec security-association ecn propogate`	Enables the Explicit Congestion Notification (ECN) settings per SA for an IPsec traffic flow in a crypto map.
Step 7	**end** **Example:** `Device(config-crypto-map)# end`	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 8	**show crypto map ipsec sa** **Example:** `Device# show crypto map ipsec sa`	Displays the settings used by IPsec SAs.

Example

The following is sample output from the **show crypto map ipsec sa** command:

```
Device# show crypto map ipsec sa

interface: Tunnel0
 Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
 PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0xF4E01B9A(4108327834)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
```

```
 conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xE963D1EC(3915633132)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

Configuration Examples for RFC 430x IPsec Support

Example: Configuring RFC 430x IPsec Support Globally

The following examples shows how to configure RFC 430x IPsec Support globally:

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec security-association dummy seconds 15
Device(config)# crypto ipsec security-association ecn propogate
Device(config-crypto-map)# exit
```

Example: Configuring RFC 430x IPsec Support Per Crypto Map

The following examples shows how to configure RFC 430x IPsec Support per crypto map:

```
Device> enable
Device# configure terminal
Device(config)# crypto map cmap 1 ipsec-isakmp
Device(config-crypto-map)# set security-association copy
Device(config-crypto-map)# set security-association dummy seconds 15
Device(config-crypto-map)# set security-association ecn propogate
Device(config-crypto-map)# end
Device# show crypto map ipsec sa

interface: Tunnel0
 Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
```

```
remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
 PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0xF4E01B9A(4108327834)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xE963D1EC(3915633132)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

Additional References for RFC 430x IPsec Support

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IKEv2 configuration	
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 4301	*Security Architecture for the Internet Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for RFC 430x IPsec Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 216: Feature Information for RFC430x IPsec Support

Feature Name	Releases	Feature Information
RFC430x IPsec Support Phase 1		The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301. The following commands were introduced or modified: **crypto ipsec security-association dummy, crypto ipsec security-association ecn, set ipsec security-association dfbit, set ipsec security-association dummy, set ipsec security-association ecn, show crypto map ipsec sa.**
RFC430x IPsec Support Phase 2		The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301. No commands were modified or updated for this feature.

PART XIV

Unified Threat Defense

CHAPTER 163

Cisco Firepower Threat Defense for ISR

Cisco Firepower Threat Defense is Cisco's premier network security option. It provides a comprehensive suite of security features, such as firewall capabilities, monitoring, alerts, and Intrusion Detection System (IDS)

This module describes how to configure and deploy IDS on Cisco Integrated Services Routers (ISRs).

Restrictions for Cisco Firepower Threat Defense for ISR

- Multicast traffic is not inspected.
- IPv6 traffic cannot be exported.

Information About Cisco Firepower Threat Defense for ISR

Cisco Firepower Threat Defense for ISR Overview

Cisco Firepower Threat Defense is a premier security solution that provides enhanced inspection for packet flows.

The Cisco Firepower Threat Defense solution consists of the following two entities:

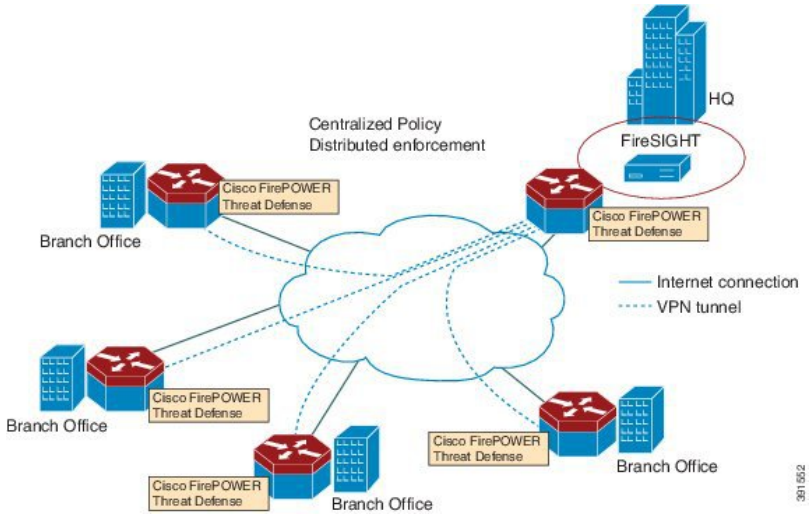
- Cisco FireSIGHT—A centralized policy and reporting entity that can run anywhere in the network. This can be the Cisco FireSIGHT appliance or a virtual installation on a server class machine.
- Virtual Firepower sensor—Security entities that implement policies, and send events and statistics back to the defense center. The Firepower sensor is hosted on Cisco Unified Computing System (UCS) E-Series Blade. Both the FireSIGHT and sensor are distributed as virtual packages.

UCS E-Series Blades are general purpose blade servers that are housed within Cisco Integrated Services Routers (ISR) Generation 2 (G2) and Cisco ISR 4000 Series Integrated Services Routers. These blades can be deployed either as bare-metal on operating systems or as virtual machines on hypervisors. There are two internal interfaces that connect a router to an UCS E-Series Blade. On ISR G2, Slot0 is a Peripheral Component Interconnet Express (PCIe) internal interface, and UCS E-Series Slot1 is a switched interface connected to the backplane Multi Gigabit Fabric (MGF). In Cisco ISR 4000 Series Routers, both internal interfaces are connected to the MGF.

A hypervisor is installed on the UCS E-Series Blade, and Cisco Firepower Threat Defense runs as a virtual machine on it. The Cisco Firepower Threat Defense OVA file is directly installed on the UCS E-Series Blade using the hypervisor operating system. Cisco Firepower Threat Defense runs as an anonymous inline device with no additional communication with the router. Traffic is diverted from the ingress physical interface to the Cisco Firepower Threat Defense that runs on the UCS E-Series Blade.

The following figure shows a Cisco Firepower Threat Defense deployment scenario. In this figure, the traffic lines between sensors and FireSIGHT are control connections. Packets are routed through these connections using router forwarding rules.

Figure 85: Cisco Firepower Threat Defense Deployment Scenario



By default, the virtualized Cisco Firepower sensor comes with three interfaces, one for management, and two others for traffic analysis. These interfaces must be mapped to the UCS E-Series interfaces.

UCS-Based Hosting

The Cisco Unified Computing System (UCS) E-Series Blade provides a generic server blade for hosting applications. This blade typically runs VMware ESXi hypervisor and is managed through vSphere like other VMWare deployments.

If the Firepower sensor is hosted on the Cisco UCS E-Series Blade, you must specify the Cisco IOS interfaces connected to Cisco Firepower Threat Defense. Applications running within the UCS E-Series Blade are only loosely coupled with Cisco IOS, and to determine the interfaces that are attached to appliances a mapping of the interfaces must be done. Interfaces to connect to the Cisco UCS E-Series Blade are Bridge Domain Interfaces (BDI).

The following Cisco UCS E-Series Blades are supported for hosting the Firepower sensor:

- UCS-E 120S
- UCS-E 140D
- UCS-E 140S
- UCS-E 160D
- UCS-E 180D

IDS Packet Flow in Cisco Firepower Threat Defense

Cisco Firepower Threat Defense supports Intrusion Detection System (IDS). In IDS mode, traffic is copied to the sensor and is analyzed for threats. IDS mode cannot enforce policies; it can detect and report violations. In IDS mode, traffic is replicated from interfaces and redirected to Cisco Firepower Threat Defense that runs on the Cisco UCS E-Series blade.

IDS copies the traffic and analyzes them for threats. Enable the **utd** command to replicate packets to the Firepower sensor based on one of the following criteria:

- If global inspection is enabled, all packets that flow through a router are replicated to the sensor.
- If per interface inspection is enabled, packets are replicated only if the input or output interface has enabled the **utd** command for inspection.

To view the interfaces that have enabled packet inspection in IDS mode, use the **show platform software utd interfaces** command. The packet replication occurs as one of the first output features.

For general packet processing, features that are applied to a packet form an ordered sequence that is determined by the configuration of the device. In general, these features are grouped as either input or output features, with the routing function marking the boundary between the two. The IDS packet replication occurs as one of the first output features and so if any input feature drops the packet, it will not be replicated to the IDS engine.

Firepower Sensor Interfaces

The Firepower sensor virtual appliance has three network interfaces—two for analyzing the traffic and one for management connectivity to FireSIGHT. The two traffic-bearing interfaces are represented as two virtual interfaces; Bridge Domain Interfaces (BDIs), in the configuration.

Although two interfaces are available for analyzing the traffic, only one traffic-bearing interface is used for Intrusion Detection System (IDS).

The Firepower sensor is connected to the management network and appears as another host on the LAN segment.

Note To monitor VLAN traffic in your virtual environment, set the VLAN ID of the promiscuous port to 4095.

Cisco Firepower Threat Defense Interoperability

Cisco Firepower Threat Defense supports Intrusion Detection System (IDS). In IDS mode, selected traffic is copied to the Firepower sensor for analysis.

Cisco Firepower Threat Defense interoperates with the following features:

- Zone-based firewall—Application layer gateways (ALGs), application inspection and controls (AICs), and policies configured between zones
- Network Address Translation (NAT)

Note Cisco Firepower Threat Defense does not support outside address translation, because there is no mechanism to inform Firepower Threat Defense about outside global addresses. However; you can still enable address translation on outside interfaces. Intrusion Prevention System (IPS) or IDS is invoked after NAT on the ingress interface, and before NAT on the egress interface, always using inside addresses.

- Crypto
- Intelligent WAN (IWAN)
- Kernel-based Virtual Machine Wide-Area Application Services (kWAAS)

Hardware and Software Requirements for Cisco Firepower Threat Defense

The following hardware is required to run the Cisco Firepower Threat Defense solution:

- Cisco Firepower Sensor version 5.4
- Cisco Integrated Services Routers (ISR) 4000 Series Routers
- Cisco Unified Computing System (UCS) E-Series Blade
- Cisco FireSIGHT

The following software is required to run the Cisco Firepower Threat Defense solution:

- UCS-E hypervisor
- ESXi 5.0.0, 5.1.0, or 5.5.0
- Cisco Firepower Sensor version Cisco IOS XE Release 3.14S and later releases
- Cisco FireSIGHT version 5.2, 5.3 or 5.4. FireSIGHT only supports the current version and is backward compatible with only the previous version. In case, your Cisco Firepower Sensor version is 5.4, then you have to use FireSIGHT version 5.4 or 5.3.

Obtaining Cisco Firepower Threat Defense License

Cisco ISR 4000 Series Integrated Services Routers must have the security K9 license and Application Experience (AppX) license to enable the Cisco Firepower Threat Defense.

```
Technology Package License Information:
------------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
```

```
-----------------------------------------------------------------
appx              appxk9            EvalRightToUse   appxk9
uc                uck9              EvalRightToUse   uck9
security          securityk9        EvalRightToUse   securityk9
ipbase            ipbasek9          Permanent        ipbasek9
```

How to Deploy Cisco Firepower Threat Defense for ISR

To deploy Cisco Firepower Threat Defense Intrusion Detection System (IDS), perform the following tasks:

1. Obtain the Firepower sensor package.
2. Install the Firepower sensor package through a hypervisor, such as VMWare VSphere.
3. Configure router interfaces for traffic redirection.
   - Bridge-Domain interface (BDI) configuration for Cisco ISR 4000 Series Routers.
   - VLAN configuration for Cisco ISR Generation 2 routers.
4. Bootstrap the Firepower sensor.
5. Configure a policy in Cisco FireSIGHT.
   - The policy is configured through the FireSIGHT GUI.
6. Enable inspection.

Obtaining the Firepower Sensor Package

To deploy the Firepower sensor on an Unified Computing System (UCS) E-Series Blade, download and save the OVA file. OVA is an Open Virtualization Archive that contains a compressed and installable version of a virtual machine. Download the OVA file from https://support.sourcefire.com/sections/1/sub_sections/51#5-2-virtual-appliances.

Installing the Firepower Sensor OVA File

Install the Firepower Sensor OVA on a UCS E-Series Blade, using a hypervisor, such as VMWare VSphere.

Installing Firepower Sensor on a UCS E-Series Blade

This section describes how to install the Firepower Sensor on a Unified Computing System (UCS) E-Series Blade that is installed on Cisco ISR 4000 Series Integrated Services Routers:

1. Install the UCS E-Series card.
2. Verify that the card is running by using the **show platform** command.
3. Configure the Cisco Integrated Management Controller (CIMC) port.

   The CIMC GUI is a web-based management interface for E-Series Servers. You can launch the CIMC GUI to manage the server from any remote host that meets the following minimum requirements:

   - Java 1.6 or later
   - HTTP or HTTPS-enabled
   - Adobe Flash Player 10 or later

   The CIMC runs on the port that is named management. The following example shows how to bootstrap the management port with an IP address:

```
ucse subslot 1/0
  imc access-port dedicated
  imc ip-address 10.66.152.158 255.255.255.0
!
```

Connect to the CIMC through the browser by using the default login and password, which are admin and password, respectively. Based on the configuration example, the browser address is https://10.66.152.158.

4. Install ESXi.

   Download the ESXi image for your Cisco UCS E-Series Blade from https://my.vmware.com/web/vmware/details?downloadGroup=CISCO-ESXI-5.1.0-GA-25SEP2012&productId=284.

5. Install Firepower Sensor by using VMWare VSphere on the Cisco UCS E-Series blade.
6. Configure traffic redirect. For more information, see the section “Configuring Traffic Redirect on Cisco UCS E-Series Blade”.
7. Configure the VMWare vSwitch. The Virtual Machine Network Interface Card (VMNIC) mapping on ISR 4000 Series Routers is as follows:

   - VMNIC0—Mapped to UCS E-Series interface x/0/0 on the router backplane
   - VMNIC1—Mapped to UCS E-Series interface x/0/1 on the router backplane
   - VMNIC2—Mapped to UCS E-Series frontplane GigabitEthernet 2 interface.
   - VMNIC3—Mapped to UCS E-Series frontplane GigabitEthernet 3 interface.

Note VMNIC3 is only available on UCS E-Series 140D, 160Dm and 180D.

UCS E-Series 120S and 140S have 3 network adaptors and one management port. UCS E-Series 140D, 160Dm and 180D have 4 network adaptors.

Configuring Traffic Redirect on Cisco UCS E-Series Blade

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address**
5. **no negotiation auto**
6. **switchport mode trunk**
7. **no mop enabled**
8. **no mop sysid**
9. **service instance** *service-instance-number ethernet*
10. **encapsulation dot1q** *vlan-id*
11. **rewrite ingress tag pop** {**1** | **2**} **symmetric**
12. **bridge domain** *bridge-ID*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ucse 1/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes an IP address or disables IP processing on an interface.
Step 5	**no negotiation auto** **Example:** `Router(config-if)# no negotiation auto`	Disables advertisement of speed, duplex mode, and flow control on an interface.
Step 6	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies a trunking VLAN Layer 2 interface.
Step 7	**no mop enabled** **Example:** `Router(config-if)# no mop enabled`	Disables the Maintenance Operation Protocol (MOP) on an interface.
Step 8	**no mop sysid** **Example:** `Router(config-if)# no mop sysid`	Disables the sending of periodic MOP system identification messages from an interface.
Step 9	**service instance** *service-instance-number ethernet* **Example:** `Router(config-if)# service instance 10 ethernet`	Configures an Ethernet service instance on an interface and enters Ethernet service-instance configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 10`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
Step 11	**rewrite ingress tag pop** {**1** | **2**} **symmetric** **Example:**	Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance.

	Command or Action	Purpose
	`Router(config-if-srv)# rewrite ingress tag pop 1 symmetric`	
Step 12	**bridge domain** *bridge-ID* **Example:** `Router(config-if-srv)# bridge domain 10`	Binds a service instance or a MAC tunnel to a bridge domain instance.
Step 13	**end** **Example:** `Router(config-if)# end`	Exits Ethernet service-instance configuration mode and returns to privileged EXEC configuration mode.

Bootstrapping the Firepower Sensor

You must configure the Firepower Sensor manually. Perform this task to configure a Firepower sensor to communicate with FireSIGHT. For more information, see https://support.sourcefire.com/sections/10.

A sensor running on a Cisco Unified Computing System (UCS) E-Series Blade is bootstrapped by logging into the console of the Firepower Sensor virtual machine through VSphere.

Note Firepower Sensor must be installed and deployed before bootstrapping it.

SUMMARY STEPS

1. Provide the default username and password to login.
2. **configure network ipv4 manual** *ip-address network-mask default-gateway*
3. **configure network dns servers** *dns-server*
4. **configure network dns searchdomains** *domain-name*
5. **configure manager add** *dc-hostname registration-key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Provide the default username and password to login.	To configure the sensor, the default username and password are admin and Sourcefire, respectively. • You must change the admin password after you login to the Firepower Sensor the first time.
Step 2	**configure network ipv4 manual** *ip-address network-mask default-gateway* **Example:** `Device# configure network ipv4 manual 10.66.152.137 255.255.255.0 10.66.152.1`	Configures network connectivity.

	Command or Action	Purpose
Step 3	**configure network dns servers** *dns-server* **Example:** `Device# configure network dns servers 192.10.26.10`	Configures domain name system (DNS) servers.
Step 4	**configure network dns searchdomains** *domain-name* **Example:** `Device# configure network dns searchdomains cisco.com`	Configures DNS search domains.
Step 5	**configure manager add** *dc-hostname registration-key* **Example:** `Device# configure manager sourcefire-dc.cisco.com cisco-sf`	Associates the sensor with the FireSIGHT. • The *registration key* is a string selected by the user that is later used to register the sensor with FireSIGHT.

Example

The following is sample output from the **show network** command that displays the configured network settings of the Firepower Sensor:

```
Device# show network

-------------------------------------------------
IPv4
Configuration             : manual
Address                   : 10.66.152.137
Netmask                   : 255.255.255.0
Gateway                   : 10.66.152.1
MAC Address               : 44:03:A7:43:05:AD
Management port           : 8305
-------------------------------------------------
IPv6
Configuration             : disabled
Management port           : 8305
-------------------------------------------------
```

The following is sample output from the **show dns** command that displays the configured DNS settings:

```
Device# show dns

search cisco.com
nameserver 192.10.26.10
```

The following is sample output from the **show managers** command that displays the configured management settings:

```
Device# show managers

Host                      : sourcefire-dc.cisco.com
Registration Key          : cisco-sf
Registration              : pending
RPC Status                :
```

Enabling IDS Inspection Globally

Based on your requirements, you can configure the Intrusion Detection System (IDS) inspection at a global level or at an interface level.

You cannot enable IDS inspection on dedicated management interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **utd enable**
4. **utd engine advanced**
5. **threat detection**
6. **exit**
7. **utd**
8. **all-interfaces**
9. **engine advanced**
10. **fail close**
11. **rate** *pps-rate*
12. **redirect-interface** *interface interface-number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**utd enable** **Example:** `Router(config)# utd enable`	Enters unified threat defense configuration mode.
Step 4	**utd engine advanced** **Example:** `Router(config)# utd engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advanced engine configuration. mode.
Step 5	**threat detection** **Example:** `Router(config-utd-eng-adv)# threat detection`	Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**utd** **Example:** `Router(config)# utd`	Enters unified threat defense configuration mode.
Step 8	**all-interfaces** **Example:** `Router(config-utd)# all-interfaces`	Configures UTD on all Layer 3 interfaces of the device
Step 9	**engine advanced** **Example:** `outer(config-utd)# engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advaned engine configuration.
Step 10	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 11	**rate** *pps-rate* **Example:** `Device(config-engine-std)# rate 2000000`	(Optional) Specify the pps rate to push to the sensor. The range is from 1000 to 4000000.
Step 12	**redirect-interface** *interface interface-number* **Example:** `Router(config-utd)# redirect-interface BDI 10`	Configures IDS traffic redirect on an interface.
Step 13	**end** **Example:** `Router(config-utd)# end`	Exits unified threat defense configuration mode and returns to privileged EXEC mode.

Enabling IDS Inspection per Interface

Based on your requirements, you can configure the Intrusion Detection System (IDS) inspection at a global level or at an interface level.

You cannot enable IDS inspection on dedicated management interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **utd enable**
5. **exit**
6. Repeat Steps 3 to 5, on all interfaces that require IDS inspection. Do not configure inspection on management interfaces.
7. **utd engine advanced**
8. **threat detection**
9. **utd**
10. **engine advanced**
11. **fail close**
12. **rate** *range*
13. **redirect interface** *type number*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**utd enable** **Example:** `Router(config-if)# utd enable`	Enables intrusion detection on an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	Repeat Steps 3 to 5, on all interfaces that require IDS inspection. Do not configure inspection on management interfaces.	-
Step 7	**utd engine advanced** **Example:** `Router(config)# utd engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advanced engine configuration. mode.
Step 8	**threat detection** **Example:**	Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.

	Command or Action	Purpose
	`Router(config-utd-eng-adv)# threat detection`	
Step 9	**utd** **Example:** `Router(config)# utd`	Enters unified threat defense configuration mode.
Step 10	**engine advanced** **Example:** `outer(config-utd)# engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advaned engine configuration.
Step 11	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 12	**rate** *range* **Example:** `Device(config-engine-std)# rate 1000`	(Optional) Specify the pps rate to push to the sensor. The range is 1000 to 4000000.
Step 13	**redirect interface** *type number* **Example:** `Router(config-utd)# redirect interface BDI 10`	Configures IDS traffic redirect on an interface.
Step 14	**end** **Example:** `Router(config-utd)# end`	Exits unified threat defense configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco Firepower Threat Defense on ISR

Example: Configuring Traffic Redirect on Cisco UCS E-Series Blade

This example shows how to configure ingress and egress interfaces for traffic redirect:

```
Router# configure terminal
Router(config)# interface ucse 1/0/0
Router(config-if)# no ip address
Router(config-if)# no negotiation auto
Router(config-if)# switchport mode trunk
Router(config-if)# no mop enabled
Router(config-if)# no mop sysid
Router(config-if)# exit
Router(config)# interface ucse 1/0/1
Router(config-if)# no ip address
Router(config-if)# no negotiation auto
```

```
Router(config-if)# switchport mode trunk
Router(config-if)# no mop enabled
Router(config-if)# no mop sysid
Router(config-if)# service instance 10 ethernet
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag pop 1 symmetric
Router(config-if-srv)# bridge domain 10
Router(config-if-srv)# exit
Router(config-if)# exit
Router(config)# interface BDI 10
Router(config-if)# no shutdown
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if-srv)# end
```

Example: Bootstrapping the Firepower Sensor

The following example shows how to bootstrap the Firepower Threat Defense sensor:

```
Sourcefire3D login: admin
Password: Sourcefire
Last login: Tue Nov 12 11:15:03 UTC 2013 on tty1

Copyright 2001-2013, Sourcefire, Inc. All rights reserved. Sourcefire is
a registered trademark of Sourcefire, Inc. All other trademarks are
property of their respective owners.

Sourcefire Linux OS v5.2.0 (build 135)
Sourcefire Virtual Device 64bit v5.2.0 (build 838)

> configure password
Enter current password:
Enter new password:
Confirm new password:

> configure network ipv4 manual 10.66.152.137 255.255.255.0 10.66.152.1
Setting IPv4 network configuration.
ADDRCONF(NETDEV_UP): eth0: link is not ready
e1000: eth0: e1000_phy_read_status: Error reading PHY register
e1000: eth0: e1000_watchdog_task: NIC Link is Up
1000 Mbps Full Duplex, Flow Control: None
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

Network settings changed.

> configure network dns servers 192.10.26.10

> configure network dns searchdomains cisco.com

configure manager add sourcefire-dc.cisco.com cisco-sf
Manager successfully configured.
```

Example: Enabling IDS Inspection Globally

```
Router# configure terminal
Router(config)# utd enable
```

```
Router(config-utd)# utd engine advanced
Router(config-utd-adv)# threat detection
Router(config-utd-adv)# exit
Router(config)# utd
Router(config-utd)# all-interfaces
Router(config-utd)# engine advanced
Router(config-utd)# fail close
Router(config-utd)# rate 1000
Router(config-utd)# redirect-interface BDI 10
Router(config-utd)# end
```

Example: Enabling IDS Inspection per Interface

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# utd enable
Router(config-utd)# utd engine advanced
Router(config-utd-adv)# threat detection
Router(config-utd-adv)# exit
Router(config)# utd
Router(config-utd)# engine advanced
Router(config-utd)# fail close
Router(config-utd)# rate 1000
Router(config-utd)# redirect-interface BDI 10
Router(config-utd)# end
```

Verifying and Monitoring IDS Inspection

Use the following commands to verify and monitor your Intrusion Detection System (IDS) deployment:

SUMMARY STEPS

1. **enable**
2. **debug platform condition feature utd controlplane**
3. **debug platform condition feature utd dataplane submode**
4. **show platform hardware qfp active utd** {**config** | **status** [**all**] [**clear**] [**drop**] [**general**]}

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 debug platform condition feature utd controlplane

Enables the debugging of the IDS configuration and status information.

Example:

```
Router# debug platform condition feature utd controlplane

network RF:
  network-rf idb-sync-history events debugging is on
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

Feature      Type           Submode       Level
-----------|-------------|------------------------------
UTD           controlplane                info

IOSXE Packet Tracing Configs:

Packet Infra debugs:

Ip Address                                                 Port
-----------------------------------------------------------|----------
```

Step 3 **debug platform condition feature utd dataplane submode**

Enables the debugging of IDS packet flow information.

Example:

```
Router# debug platform  condition feature utd dataplane submode

network RF:
  network-rf idb-sync-history events debugging is on
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

Feature      Type           Submode                    Level
-----------|-------------|---------------------|----------
UTD           controlplane                          info
UTD           dataplane     fia proxy punt          info

IOSXE Packet Tracing Configs:

Packet Infra debugs:

Ip Address                                                 Port
-----------------------------------------------------------|----------
```

Step 4 **show platform hardware qfp active utd** {**config** | **status** [**all**] [**clear**] [**drop**] [**general**]}

Displays information about the IDS inspection in the Cisco Quantum Flow Processor (QFP).

Example:

```
Router# show platform hardware qfp active utd config

Global flags: 0x40004
Num divert interfaces: 1
Divert UIDBs: 65521 0
FIB information
[0][0] 0x309e3c30
[0][1] 0x0
[1][0] 0x309e4040
```

```
[1][1] 0x0
```

Additional References for Cisco Firepower Threat Defense for ISR

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
UCS E-Series Servers	http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/e/2-0/gs/guide/b_2_0_Getting_S

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Cisco Firepower Threat Defense for ISR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 217: Feature Information for Cisco Firepower Threat Defense for ISR

Feature Name	Releases	Feature Information

CHAPTER 164

Snort IPS

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the open source Snort solution to enable IPS and IDS. The Snort IPS feature is available in Cisco IOS XE Release 3.16.1S, 3.17S, and later releases.

Note The Virtual Routing and Forwarding (VRF) feature is supported on Snort IPS configuration from Cisco IOS XE Denali Release 16.3.1 and later releases.

This module explains the feature and how it works.

Restrictions for Snort IPS

The following restrictions apply to the Snort IPS feature:

- When you enable boost license on Cisco 4000 Series ISRs, you cannot configure the virtual-service container for Snort IPS.
- Incompatible with the Zone-Based Firewall SYN-cookie feature.
- Network Address Translation 64 (NAT64) is not supported.
- SnortSnmpPlugin is required for SNMP polling in open source Snort. Snort IPS does not support SNMP polling capabilities or MIBs as the SnortSnmp plugin is not installed on UTD.

- **IOS syslog is rate limited and as a result, all alerts generated by Snort may not be visible via the IOS Syslog. However, you can view all Syslog messages if you export them to an external log server.**

Information About Snort IPS

Snort IPS Overview

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the Snort engine to provide IPS and IDS functionalities.

Snort is an open source network IPS that performs real-time traffic analysis and generates alerts when threats are detected on IP networks. It can also perform protocol analysis, content searching or matching, and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, and so on. The Snort engine runs as a virtual container service on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series.

The Snort IPS feature works in the network intrusion detection and prevention mode that provides IPS or IDS functionalities. In the network intrusion detection and prevention mode, Snort performs the following actions:

- Monitors network traffic and analyzes against a defined rule set.
- Performs attack classification.
- Invokes actions against matched rules.

Based on your requirements, you can enable Snort either in IPS or IDS mode. In IDS mode, Snort inspects the traffic and reports alerts, but does not take any action to prevent attacks. In IPS mode, in addition to intrusion detection, actions are taken to prevent attacks.

The Snort IPS monitors the traffic and reports events to an external log server or the IOS syslog. Enabling logging to the IOS syslog may impact performance due to the potential volume of log messages. External third-party monitoring tools, which supports Snort logs, can be used for log collection and analysis.

Snort IPS Signature Package

The UTD OVA is included in the security license of the router. By default, the router is loaded only with community signature package. There are two types of subscriptions :

- Community Signature Package
- Subscriber-based Signature Package

The community signature package rule set offers limited coverage against threats. The subscriber-based signature package rule set offers the best protection against threats. It includes coverage in advance of exploits, and also provides the fastest access to the updated signatures in response to a security incident or the proactive discovery of a new threat. This subscription is fully supported by Cisco and the package will be updated on Cisco.com. You can download the subscriber-based signature package from the Download Software page.

If the user downloads the signature package manually from the download software page, then the user should ensure that the package has the same version as the Snort engine version. For example, if the Snort engine version is 2982, then the user should download the same version of the signature package. If there is a version mismatch, the signature package update will be rejected and it will fail.

Note When the signature package is updated, the engine will be restarted and the traffic will be interrupted or bypass inspection for a short period depending on their data plane fail-open/fail-close configuration.

Minimum Supported Cisco IOS XE Release and UTD Package Versions for Signature Updates

Table 1 below lists the minimum Cisco IOS XE releases and their respective UTD package versions that support signature package updates post January, 2020. The Cisco IOS XE releases and their respective UTD package versions that are prior to those listed in the table are not supported. The Cisco IOS XE releases and their respective UTD package versions that are more recent than those listed in the table are supported from their first release.

Table 218: UTD Signature Package Update Support Version Matrix

Cisco IOS XE Release	UTD Package Version
16.6.7	1.0.10_SV29111_XE_16_6
16.9.4	1.0.4_SV29111_XE_16_9
16.10.2	1.0.9_SV2.9.11.1_XE16.10

Note When UTD is oversubscribed, the threat defence channel state changes between green and red. The UTD dataplane either drops all further packets if fail-close is configured or forwards the packets un-inspected if fail-close is not configured (default). When the UTD serviceplane recovers from over-subscription, it responds to the UTD dataplane with the green status.

Snort IPS Solution

The Snort IPS solution consists of the following entities:

- Snort sensor—Monitors the traffic to detect anomalies based on the configured security policies (that includes signatures, statistics, protocol analysis, and so on) and sends alert messages to the Alert/Reporting server. The Snort sensor is deployed as a virtual container service on the router.
- Signature store—Hosts the Cisco Signature packages that are updated periodically. These signature packages are downloaded to Snort sensors either periodically or on demand. Validated signature packages are posted to Cisco.com. Based on the configuration, signature packages can be downloaded from Cisco.com or a local server.

The following domains are accessed by the router in the process of downloading the signature package from cisco.com:

- api.cisco.com
- apx.cisco.com
- cloudsso.cisco.com
- cloudsso-test.cisco.com
- cloudsso-test3.cisco.com
- cloudsso-test4.cisco.com
- cloudsso-test5.cisco.com
- cloudsso-test6.cisco.com
- cloudsso.cisco.com
- download-ssc.cisco.com
- dl.cisco.com
- resolver1.opendns.com
- resolver2.opendns.com

Note If you are downloading signature packages from a local server to hold the signature packages, only HTTP is supported.

Signature packages must be manually downloaded from Cisco.com to the local server by using Cisco.com credentials before the Snort sensor can retrieve them.

The Snort container performs a domain-name lookup (on the DNS server(s) configured on the router) to resolve the location for automatic signature updates from Cisco.com or on the local server, if the URL is not specified as the IP address.

- Alert/Reporting server—Receives alert events from the Snort sensor. Alert events generated by the Snort sensor can either be sent to the IOS syslog or an external syslog server or to both IOS syslog and external syslog server. No external log servers are bundled with the Snort IPS solution.
- Management—Manages the Snort IPS solution. Management is configured using the IOS CLI. Snort Sensor cannot be accessed directly, and all configuration can only be done using the IOS CLI.

Overview of Snort Virtual Service Interfaces

The Snort sensor runs as a service on routers. Service containers use virtualization technology to provide a hosting environment on Cisco devices for applications.

You can enable Snort traffic inspection either on a per interface basis or globally on all supported interfaces. The traffic to be inspected is diverted to the Snort sensor and injected back. In Intrusion Detection System (IDS), identified threats are reported as log events and allowed. However, in Intrusion Prevention System (IPS), action is taken to prevent attacks along with log events.

The Snort sensor requires two VirtualPortGroup interfaces. The first VirtualPortGroup interface is used for management traffic and the second for data traffic between the forwarding plane and the Snort virtual container service. Guest IP addresses must be configured for these VirtualPortGroup interfaces. The IP subnet assigned to the management VirtualPortGroup interface should be able to communicate with the Signature server and Alert/Reporting server.

The IP subnet of the second VirtualPortGroup interface must not be routable on the customer network because the traffic on this interface is internal to the router. Exposing the internal subnet to the outside world is a security risk. We recommend the use of 192.0.2.0/30 IP address range for the second VirtualPortGroup subnet. The use of 192.0.2.0/24 subnet is defined in RFC 3330.

You can also use the management interface under the **virtual-service** command for management traffic. If you configure the management interface, you still need two VirtualPortGroup interfaces. However, do not configure the **guest ip address** for the first VirtualPortGroup interface.

You can assign the Snort virtual container service IP address on the same management network as the router on which the virtual service is running. This configuration helps if the syslog or update server is on the management network and is not accessible by any other interfaces.

Virtual Service Resource Profile

The Snort IPS virtual service supports three resource profiles: Low, Medium, and High. These profiles indicate the CPU and memory resources required to run the virtual service. You can configure one of these resource profiles. The resource profile configuration is optional. If you do not configure a profile, the virtual service is activated with its default resource profile. This table provides the resource profiles details for Cisco 4000 Series ISR and Cisco Cloud Services Router 1000v Series.

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4321 ISR	Default	50%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
Cisco 4331 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)

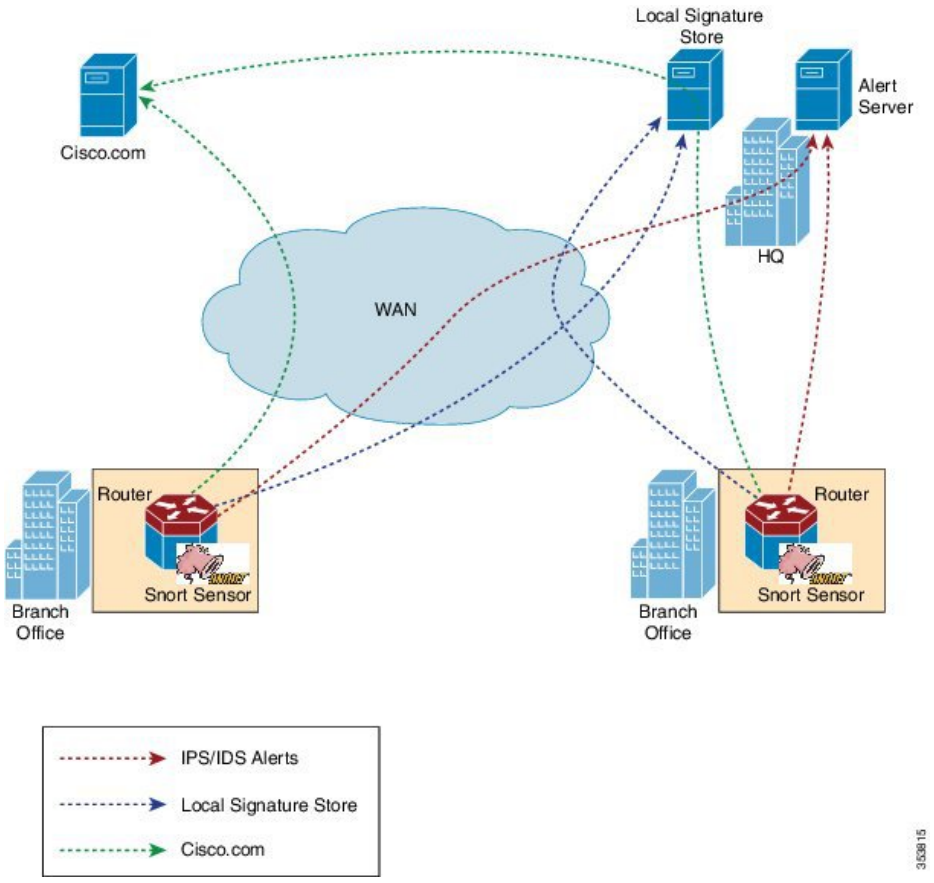
Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4351 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
Cisco 4431 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)
Cisco 4451 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco CSR 1000V	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 3GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)

Deploying Snort IPS

The figure illustrates a Snort IPS deployment scenario:

Figure 86: Snort IPS Deployment Scenario



The following steps describes the deployment of the Snort IPS solution:

- The Snort OVA file is copied to Cisco routers, installed, and then activated.
- Signature packages are downloaded either from Cisco.com or a configured local server to Cisco routers.
- Network intrusion detection or prevention functionality is configured.
- The Alert/Reporting server is configured to receive alerts from the Snort sensor.

How to Deploy Snort IPS

To deploy Snort IPS on supported devices, perform the following tasks:

1. Provision the device.

   Identify the device to install the Snort IPS feature.

2. Obtain the license.

   The Snort IPS functionality is available only in Security Packages which require a security license to enable the service. This feature is available in Cisco IOS XE Release 3.16.1S, 3.17S, and later releases.

Note Contact Cisco Support to obtain the license.

3. Install the Snort OVA file.
4. Configure VirtualPortGroup interfaces and virtual-service.
5. Activate the Snort virtual container service.
6. Configure Snort IPS or IDS mode and policy.
7. Configure the reporting of events to an external alert/log server or IOS syslog or both.
8. Configure the Signature update method.
9. Update the Signatures.
10. Enable IPS globally or on desired interfaces.

Installing the Snort OVA File

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. The Snort IPS is available as a virtual container service. You must download this OVA file on to the router and use the **virtual-service install** CLI to install the service.

The service OVA file is not bundled with the Cisco IOS XE Release images that are installed on the router. However, the OVA files may be preinstalled in the flash of the router.

You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

SUMMARY STEPS

1. **enable**
2. **virtual-service install name** *virtual-service-name* **package** *file-url* **media** *file-system*
3. **show virtual-service list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**virtual-service install name** *virtual-service-name* **package** *file-url* **media** *file-system* **Example:** `Device# virtual-service install name UTDIPS package harddisk:utd-ips-v102.ova media harddisk:`	Installs an application on the virtual services container of a device. • The length of the name is 20 characters. Hyphen (-) is not a valid character. • You must specify the complete path of the OVA package to be installed. **Note** OVA installation works on both hard disk and bootflash, the preferred filesystem to install the OVA will be hard disk.
Step 3	**show virtual-service list** **Example:** `Device# show virtual-service list`	Displays the status of the installation of all applications installed on the virtual service container.

Configuring VirtualPortGroup Interfaces and Virtual Service

You must configure two VirtualPortGroup interfaces and configure guest IP addresses for both interfaces. However, if you configure a management interface by using the **vnic management GigabitEthernet0** command, then do not configure the guest IP address for the first VirtualPortGroup interface.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

Note Before you change the Cisco IOS software image from any of the XE 3.x versions to XE 16.2.1, or from XE 16.2.1 to any of the XE 3.x versions, uninstall the virtual-service by using the **virtual-service uninstall name [name]** command for each virtual-service on the device. If one of the virtual-services is the ISR-WAAS service, which is installed with the **service waas enable** command, use the **service waas disable** command.

After the device is upgraded with the new version of Cisco IOS software image, re-install the virtual-services. For ISR-WAAS, use the **service wass enable** command, and for other virtual-services, use the **virtual-service install name [name] package [.ova file]** command.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *VirtualPortGroup number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **exit**
9. **virtual-service** *name*
10. **profile** *profile-name*
11. **vnic gateway VirtualPortGroup** *interface-number*
12. **guest ip address** *ip-address*
13. **exit**
14. **vnic gateway VirtualPortGroup** *interface-number*
15. **guest ip address** *ip-address*
16. **exit**
17. **vnic management GigabitEthernet0**
18. **guest ip address** *ip-address*
19. **exit**
20. **activate**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *VirtualPortGroup number* **Example:** `Device(config)# interface VirtualPortGroup 0`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. This interface is used for management traffic when the management interface GigabitEthernet0 is not used.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.252`	Sets a primary IP address for an interface. This interface needs to be routable to the signature update server and external log server.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. • This interface is used for data traffic.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.1 255.255.255.252`	Sets a primary IP address for an interface. • This IP address should not be routable to the outside network. • The IP address is assigned from the recommended 192.0.2.0/30 subnet.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**virtual-service** *name* **Example:** `Device(config)# virtual-service UTDIPS`	Configures a virtual container service and enters virtual service configuration mode. • The *name* argument is the logical name that is used to identify the virtual container service.

Step 10	**profile** *profile-name*
	Example:
	`Device(config-virt-serv)#profile high`
	Example:
	`Device(config-virt-serv)#profile multi-tenancy`
		(Optional) Configures a resource profile. If you do not configure the resource profile, the virtual service is activated with its default resource profile. The options are: low, medium, high, and multi-tenancy. (For multi-tenancy mode (Cisco CSR 1000v only), a `profile multi-tenancy` command must be configured.)

Step 11	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	Creates a virtual network interface card (vNIC) gateway interface for the virtual container service, maps the vNIC gateway interface to the virtual port group, and enters the virtual-service vNIC configuration mode. • The interface referenced in this command must be the one configured in Step 3. This command maps the interface that is used for management purposes.
Step 12	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 10.1.1.2`	(Optional) Configures a guest vNIC address for the vNIC gateway interface. • **Note** Configure this command only if the **vnic management gigabitethernet0** command specified in Step 17 is not configured.
Step 13	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.

	Command or Action	Purpose
Step 14	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 1`	Creates a vNIC gateway interface for the virtual container service, maps the vNIC gateway interface to the virtual port group, and enters the virtual-service vNIC configuration mode. • This interface referenced in this command must be the one configured in Step 6. This command maps the interface in the virtual container service that is used by Snort for monitoring the user traffic.
Step 15	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 192.0.2.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 16	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 17	**vnic management GigabitEthernet0** **Example:** `Device(config-virt-serv)# vnic management GigabitEthernet0`	(Optional) Configures the GigabitEthernet interface as the vNIC management interface. • The management interface must either be a VirtualPortGroup interface or GibagitEthernet0 interface. • If you do not configure the **vnic management GigabitEthernet0** command, then you must configure the **guest ip address** command specified in Step 12.
Step 18	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 209.165.201.1`	(Optional) Configures a guest vNIC address for the vNIC management interface and it must be in the same subnet as the management interface and GigabitEthernet0 configuration.
Step 19	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 20	**activate** **Example:** `Device(config-virt-serv)# activate`	Activates an application installed in a virtual container service.
Step 21	**end** **Example:** `Device(config-virt-serv)# end`	Exits virtual service configuration mode and returns to privileged EXEC mode.

Configuring Snort IPS Globally

Based on your requirements, configure the Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) inspection at a global level or at an interface. Perform this task to configure IPS globally on a device.

Note The term global refers to Snort IPS running on all supported interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **utd threat-inspection whitelist**
4. **generator id** *generator-id* **signature id** *signature-id* [**comment** *description*]
5. **exit**
6. **utd engine standard**
7. **logging** {**host** *hostname* | **syslog**}
8. **threat-inspection**
9. **threat** {**detection** | **protection** }
10. **policy** {**balanced** | **connectivity** | **security**}
11. **whitelist**
12. **signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute*
13. **signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]]
14. **logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
15. **exit**
16. **utd**
17. **redirect interface virtualPortGroup** *interface-number*
18. **all-interfaces**
19. **engine standard**
20. **fail close**
21. **exit**
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**utd threat-inspection whitelist** **Example:** `Device(config)# utd threat-inspection whitelist`	(Optional) Enables the UTD allowed list configuration mode.
Step 4	**generator id** *generator-id* **signature id** *signature-id* [**comment** *description*] **Example:** `Device(config-utd-whitelist)# generator id 24 signature id 24245 comment traffic from branchoffice1`	Configures signature IDs to appear in the allowed list. • Signature IDs can be copied from alerts that needs to be suppressed. • You can configure multiple signature IDs. • Repeat this step for each signature ID that needs to be added to the allowed list.
Step 5	**exit** **Example:** `Device(config-utd-whitelist)# exit`	Exits UTD allowed list configuration mode and returns to global configuration mode.
Step 6	**utd engine standard** **Example:** `Device(config)# utd engine standard`	Configures the unified threat defense (UTD) standard engine and enters UTD standard engine configuration mode.
Step 7	**logging** {**host** *hostname* | **syslog**} **Example:** `Device(config-utd-eng-std)# logging host syslog.yourcompany.com`	Enables the logging of emergency messages to a server.
Step 8	**threat-inspection** **Example:** `Device(config-utd-eng-std)# threat-inspection`	Configures threat inspection for the Snort engine.
Step 9	**threat** {**detection** | **protection** } **Example:** `Device(config-utd-eng-std-insp)# threat protection`	Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine. • The default is **detection**. • Configure the **detection** keyword to configure Intrusion Detection System (IDS).
Step 10	**policy** {**balanced** | **connectivity** | **security**} **Example:** `Device(config-utd-eng-std-insp)# policy security`	Configures the security policy for the Snort engine. • The default policy option is **balanced**.
Step 11	**whitelist** **Example:** `Device(config-utd-eng-std-insp)# whitelist`	(Optional) Enables allowed listing under the UTD engine.

	Command or Action	Purpose
Step 12	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 13	**signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-eng-std-insp)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use Cisco.com for signature updates, you must provide the username and password. If you use local server for signature updates, based on the server settings you can provide the username and password.
Step 14	**logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**} **Example:** `Device(config-utd-eng-std-insp)# logging level emerg`	Enables the log level.
Step 15	**exit** **Example:** `Device(config-utd-eng-std-insp)# exit`	Exits UTD standard engine configuration mode and returns to global configuration mode.
Step 16	**utd** **Example:** `Device(config)# utd`	Enables unified threat defense (UTD) and enters UTD configuration mode.
Step 17	**redirect interface virtualPortGroup** *interface-number* **Example:** `Device(config-utd)# redirect interface virtualPortGroup 1`	(Optional) Redirects to a VirtualPortGroup interface. This is the data traffic interface. If you do not configure this interface, it is auto-detected.
Step 18	**all-interfaces** **Example:** `Device(config-utd)# all-interfaces`	Configures UTD on all Layer 3 interfaces of the device.
Step 19	**engine standard** **Example:** `Device(config-utd)# engine standard`	Configures the Snort-based unified threat defense (UTD) engine and enters standard engine configuration mode.
Step 20	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.

	Command or Action	Purpose
Step 21	**exit** **Example:** `Device(config-eng-std)# exit`	Exits standard engine configuration mode and returns to global configuration mode.
Step 22	**end** **Example:** `Device(config-utd)# end`	Exits UTD configuration mode and returns to global configuration mode.

Configuring Snort IDS Inspection Globally

Based on your requirements, configure either Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) inspection at a global level or at an interface level. Perform this task to configure IDS on a per-interface basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **utd enable**
5. **exit**
6. Repeat Steps 3 to 5, on all interfaces that require inspection.
7. **utd threat-inspection whitelist**
8. **generator id** *generator-id* **signature id** *signature-id* [**comment** *description*]
9. **exit**
10. **utd engine standard**
11. **logging** {**host** *hostname* | **syslog**}
12. **threat-inspection**
13. **threat** {**detection** | **protection** }
14. **policy** {**balanced** | **connectivity** | **security**}
15. **whitelist**
16. **signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute*
17. **signature update server** {**cisco** | **url** *url*} [**username** *username* [**password** *password*]]
18. **logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
19. **exit**
20. **utd**
21. **redirect interface virtualPortGroup** *interface-number*
22. **engine standard**
23. **fail close**
24. **exit**
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**utd enable** **Example:** `Device(config-if)# utd enable`	Enables unified threat defense (UTD).
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	Repeat Steps 3 to 5, on all interfaces that require inspection.	–
Step 7	**utd threat-inspection whitelist** **Example:** `Device(config)# utd threat-inspection whitelist`	(Optional) Enables the UTD allowed list configuration mode.
Step 8	**generator id** *generator-id* **signature id** *signature-id* [**comment** *description*] **Example:** `Device(config-utd-whitelist)# generator id 24 signature id 24245 comment traffic from branchoffice1`	Configures signature IDs to appear on the allowed list. • Signature IDs can be copied from alerts that needs to be suppressed. • You can configure multiple signature IDs. • Repeat this step for each signature ID that needs to appear on the allowed list.
Step 9	**exit** **Example:** `Device(config-utd-whitelist)# exit`	Exits UTD allowed list configuration mode and returns to global configuration mode.
Step 10	**utd engine standard** **Example:** `Device(config)# utd engine standard`	Configures the unified threat defense (UTD) standard engine and enters UTD standard engine configuration mode.
Step 11	**logging** {**host** *hostname* | **syslog**} **Example:**	Enables the logging of critical messages to the IOSd syslog.

	Command or Action	Purpose
	`Device(config-utd-eng-std)# logging syslog`	
Step 12	**threat-inspection** **Example:** `Device(config-utd-eng-std)# threat-inspection`	Configures threat inspection for the Snort engine.
Step 13	**threat** {**detection** | **protection** } **Example:** `Device(config-utd-eng-std-insp)# threat detection`	Configures threat protection or Intrusion Detection System (IDS) as the operating mode for the Snort sensor. • Configure the **protection** keyword to configure Intrusion Prevention System (IPS).
Step 14	**policy** {**balanced** | **connectivity** | **security**} **Example:** `Device(config-utd-eng-std-insp)# policy balanced`	Configures the security policy for the Snort sensor.
Step 15	**whitelist** **Example:** `Device(config-utd-eng-std-insp)# whitelist`	(Optional) Enables allowed listing of traffic.
Step 16	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 17	**signature update server** {**cisco** | **url** *url*} [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-eng-std-insp)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use Cisco.com for signature updates, you must provide the username and password. If you use local server for signature updates, based on the server settings you can provide the username and password.
Step 18	**logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**} **Example:** `Device(config-utd-eng-std-insp)# logging level crit`	Enables the log level.
Step 19	**exit** **Example:** `Device(config-utd-eng-std-insp)# exit`	Exits UTD standard engine configuration mode and returns to global configuration mode.
Step 20	**utd** **Example:** `Device(config)# utd`	Enables unified threat defense (UTD) and enters UTD configuration mode.

	Command or Action	Purpose
Step 21	**redirect interface virtualPortGroup** *interface-number* **Example:** `Device(config-utd)# redirect interface virtualPortGroup 1`	(Optional) Redirects to a VirtualPortGroup interface. This is the data traffic interface. If you do not configure this interface, it is auto-detected.
Step 22	**engine standard** **Example:** `Device(config-utd)# engine standard`	Configures the Snort-based unified threat defense (UTD) engine and enters standard engine configuration mode.
Step 23	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 24	**exit** **Example:** `Device(config-eng-std)# exit`	Exits standard engine configuration mode and returns to global configuration mode.
Step 25	**end** **Example:** `Device(config-utd)# end`	Exits configuration mode and returns back to exec mode.

Displaying the List of Active Signatures

Active signatures are the ones that prompt Snort IDS/IPS to take action against threats. If the traffic matches with any of the active signatures, Snort container triggers alert in the IDS mode, and drops the traffic in the IPS mode.

The **utd threat-inspection signature active-list write-to bootflash: file name** command provides a list of active signatures and a summary of the total number of active signatures, drop signatures, and alert signatures.

Configuring Quality of Service Policy for Monitoring the Container's Health

It is recommended to configure a Quality of Service (QoS) policy to ensure the health probes that monitor the container's health are not impacted at high traffic rates.

SUMMARY STEPS

1. **ip access-list extended** {acl-name | acl-number}
2. sequence-number permit protocol source *source-wildcard destination destination-wildcard* [precedence] [tos *tos* tos] [log] [time-range*time-range-name*] [fragments]
3. **exit**
4. class-map { [type inspect match-all] | [match-any] } *class-map-name*
5. match access-group { *access-group* | name *access-group-name*}
6. **exit**

7. policy-map *policy-map-name*
8. class {*class-name* | class-default
9. priority level *level*
10. **exit**
11. **interface** *type number*
12. service-policy [history | {output} *policy-map-name* | type control *control-policy-name*]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip access-list extended** {acl-name | acl-number} **Example:** `Device(config)# ip access-list extended health_probes_accesslist`	Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list.
Step 2	sequence-number permit protocol source *source-wildcard destination destination-wildcard* [precedence] [tos *tos*] [log] [time-range*time-range-name*] [fragments] **Example:** `Device(config-ext-nacl)# 10 permit udp any eq 3367 any eq 3367`	Specifies a permit statement in named IP access list mode. This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need.
Step 3	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits extended ACL configuration mode and returns to global configuration mode.
Step 4	class-map { [type inspect match-all] | [match-any] } *class-map-name* **Example:** `Device(config)# class-map match-all health_probes_cmap`	Specifies the name of the class map to be created and enters QoS class map configuration mode.
Step 5	match access-group { *access-group* | name *access-group-name*} **Example:** `Device(config-cmap)# match access-group name health_probes_accesslist`	Configure the match criteria for a class map to be successful match criteria for all packets.
Step 6	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
Step 7	policy-map *policy-map-name* **Example:** `Device(config)# policy-map health_probes_pmap`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters QoS policy-map configuration mode.

	Command or Action	Purpose
Step 8	class {*class-name* | class-default **Example:** `Device(config-pmap)# class health_probes_cmap`	Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy, and enters policy-map class configuration mode.
Step 9	priority level *level* **Example:** `Device(config-pmap)# priority level 1`	Assigns priority to a traffic class at the priority level specified. • Enter the level of priority assigned to the priority class. Valid values are 1 (high priority) and 2 (low priority). The default is 1.
Step 10	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. • This interface is used for data traffic.
Step 12	service-policy [history | {output} *policy-map-name* | type control *control-policy-name*] **Example:** `Device(config-if)# service-policy output health_probes_pmap`	Attaches a policy map to a class. The name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Snort IPS

Example: Configuring VirtualPortGroup Interfaces and Virtual Service

```
Device# configure terminal
Device(config)# interface VirtualPortGroup 0
Device(config-if)# ip address 10.1.1.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup 1
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# vnic gateway VirtualPortGroup 0
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup 1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
```

```
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic management GigabitEthernet0
Device(config-virt-serv-vnic)# guest ip address 209.165.201.1
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate
Device(config-virt-serv-vnic)# end
```

Example: Configuring a Different Resource Profile

```
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# no activate
Device(config-virt-serv)# end
Device# virtual-service uninstall name UTDIPS
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# profile medium
Device(config-virt-serv)# end
Device# virtual-service install name UTDIPS package:utd.ova
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# activate
Device(config-virt-serv)# end
```

Example: Configuring Snort IPS Globally

The following example shows how to configure Intrusion Prevention System (IPS) globally on a device:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# threat protection
Device(config-utd-eng-std-insp)# policy security
Device(config-utd-eng-std)# exit
Device(config)# utd
Device(config-utd)# all-interfaces
Device(config-utd)# engine standard
Device(config-utd-whitelist)# end
Device#
```

Example: Configuring Snort IPS Inspection per Interface

The following example shows how to configure Snort Intrusion Detection System (IDS) on a per-interface basis:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# threat detection
Device(config-utd-eng-std-insp)# policy security
```

```
Device(config-utd-eng-std)# exit
Device(config)# utd
Device(config-utd)# engine standard
Device(config-eng-std)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# utd enable
Device(config-if)# exit
```

Example: Configuring UTD with VRF on both Inbound and Outbound Interface

```
Device# configure terminal
Device(config)# vrf definition VRF1
Device(config-vrf)# rd 100:1
Device(config-vrf)# route-target export 100:1
Device(config-vrf)# route-target import 100:1
Device(config-vrf)# route-target import 100:2
!
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
!
Device(config-vrf-af)# vrf definition VRF2
Device(config-vrf)# rd 100:2
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# route-target import 100:1
!
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
!
Device(config-vrf)# interface VirtualPortGroup0
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
!
Device(config-if)# interface VirtualPortGroup1
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
!
Device(config-if)# interface GigabitEthernet0/0/2
Device(config-if)# vrf forwarding VRF1
Device(config-if-vrf)# ip address 192.1.1.5 255.255.255.0
Device(config-if-vrf)# ipv6 address A000::1/64
!
Device(config-if)# interface GigabitEthernet0/0/3
Device(config-if)# vrf forwarding VRF2
Device(config-if-vrf)# ip address 192.1.1.5 255.255.255.0
Device(config-if-vrf)# ipv6 address B000::1/64
!
Device(config-if-vrf)# router bgp 100
Device(config-if-vrf)# bgp log-neighbor-changes
!
Device(config-vrf)# address-family ipv4 vrf VRF1
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
```

```
!
Device(config-vrf)# address-family ipv6 vrf VRF1
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv4 vrf VRF2
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6 vrf VRF2
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config)# utd
Device(config-utd)# all-interfaces
Device(config-utd)# engine standard
Device(config-utd)# exit

Device(config)# utd engine standard
Device(config-utd-eng-std)# logging syslog
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)# threat protection
Device(config-utd-engstd-insp)# policy security
Device(config-utd-engstd-insp)# exist
Device(config-utd-eng-std)# exit
!
Device(config)# virtual-service utd
Device(config-virt-serv)# profile low
Device(config-virt-serv)# vnic gateway VirtualPortGroup0
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.6
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate

UTD Snort IPS Drop Log
============================
2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**]
[1:30561:1] BLACKLIST DNS request for known malware
domain domai.ddns2.biz - Win.Trojan.Beebone [**]
[Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
```

Example: Configuring Logging IOS Syslog

The following example shows how to configure logging IOS syslog with the log levels on a device:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# logging syslog
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)# logging level debug
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Logging to Centralized Log Server

The following example shows how to configure logging to a centralized log server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std-insp)# logging host syslog.yourcompany.com
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# logging level info
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Signature Update from a Cisco Server

The following example shows how to configure the signature update from a Cisco server :

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update server cisco username CCOuser password
passwd123
Device(config-utd-eng-std-insp)# end
Device#
```

Note Ensure that the DNS is configured to download signatures from the Cisco server.

Example: Configuring Signature Update from a Local Server

The following example shows how to configure the signature update from a local server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update server url http://192.168.1.2/sig-1.pkg
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Automatic Signature Update

The following example shows how to configure the automatic signature update on a server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0
Device(config-utd-eng-std-insp)# signature update server cisco username abcd password
cisco123
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Performing Manual Signature Update

The following examples show how to perform a manual signature update in different ways:

```
Device# utd threat-inspection signature update
```

It takes the existing server configuration to download from or the explicit server information configured with it. These commands perform a manual signature update with the below settings:

```
Device# show utd engine standard threat-inspection signature update status

Current signature package version: 2983.4.s
Current signature package name: UTD-STD-SIGNATURE-2983-4-S.pkg
Previous signature package version: 29.0.c
----------------------------------------
Last update status: Successful
----------------------------------------
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last successful update method: Manual
Last successful update server: cisco
Last successful update speed: 3022328 bytes in 25 secs
----------------------------------------
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update method: Manual
Last failed update server: cisco
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service hnot
known'))
----------------------------------------
Last attempted update time: Mon Aug  7 02:02:32 2017 UTC
Last attempted update method: Manual
Last attempted update server: cisco
----------------------------------------
Total num of updates successful: 1
Num of attempts successful: 1
Num of attempts failed: 3
Total num of attempts: 4
----------------------------------------
Next update scheduled at: None
----------------------------------------
Current status: Idle


Device# utd threat-inspection signature update server cisco username ccouser password
passwd123

Device# utd threat-inspection signature update server url http://192.168.1.2/sig-1.pkg
```

Example: Configuring Signature Allowed Lists

The following example shows how to configure signature allowed list:

```
Device# configure terminal
Device(config)# utd threat-inspection whitelist
Device(config-utd-whitelist)# utd-whitelist)# generator id 1 signature id 23456 comment
"traffic from client x"
Device(config-utd-whitelist)# exit
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# whitelist
Device(config-utd-eng-std-insp)# end
Device#
```

Note After the allowed list signature ID is configured, Snort will allow the flow to pass through the device without any alerts and drops.

Examples for Displaying Active Signatures

Example: Displaying Active Signatures List With Connectivity Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_connectivity
Device# more bootflash:siglist_connectivity
================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Connectivity
Total no. of active signatures: 581
Total no. of drop signatures: 452
Total no. of alert signatures: 129

For more details of each signature please go to www.snort.org/rule_docs to lookup
================================================================================
List of Active Signatures:
--------------------------
<snipped>
```

Example: Displaying Active Signatures List With Balanced Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_balanced
Device# more bootflash:siglist_balanced
================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Balanced
Total no. of active signatures: 7884
Total no. of drop signatures: 7389
Total no. of alert signatures: 495

For more details of each signature please go to www.snort.org/rule_docs to lookup
================================================================================

List of Active Signatures:
--------------------------
<snipped>
```

Example: Displaying Active Signatures List With Security Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_security
Device# more bootflash:siglist_security
================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Security
Total no. of active signatures: 11224
Total no. of drop signatures: 10220
Total no. of alert signatures: 1004

For more details of each signature please go to www.snort.org/rule_docs to lookup
```

```
================================================================================

List of Active Signatures:
--------------------------
<snipped>
```

Verifying the Integrated Snort IPS Configuration

Use the following commands to troubleshoot your configuration.

SUMMARY STEPS

1. **enable**
2. **show virtual-service list**
3. **show virtual-service detail**
4. **show service-insertion type utd service-node-group**
5. **show service-insertion type utd service-context**
6. **show utd engine standard config**
7. **show utd engine standard status**
8. **show utd engine standard threat-inspection signature update status**
9. **show utd engine standard logging events**
10. **clear utd engine standard logging events**
11. **show platform hardware qfp active feature utd config**
12. **show platform software utd global**
13. **show platform software utd interfaces**
14. **show platform hardware qfp active feature utd stats**
15. **show utd engine standard statistics daq all**

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 show virtual-service list

Displays the status of the installation of all applications on the virtual service container.

Example:

```
Device# show virtual-service list

Virtual Service List:

Name                    Status             Package Name
--------------------------------------------------------------------------------
```

```
UTDIPS                    Activated          utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Step 3 **show virtual-service detail**

Displays the resources used by applications installed in the virtual services container of a device.

Example:

```
Device# show virtual-service detail


Device#show virtual-service detail
Virtual service UTDIPS detail
  State                  : Activated
  Owner                  : IOSd
  Package information
    Name                 : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path                 : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name               : UTD-Snort-Feature
      Installed version  : 1.0.1_SV2982_XE_16_3
      Description        : Unified Threat Defense
    Signing
      Key type           : Cisco development key
      Method             : SHA-1
    Licensing
      Name               : Not Available
      Version            : Not Available

  Detailed guest status

------------------------------------------------------------------------
Process               Status            Uptime           # of restarts
------------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk                 : 736 MB
    Memory               : 1024 MB
    CPU                  : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
```

```
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                          Next Hop                            Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                             48.0.0.1                           eth2
0.0.0.0/0                             47.0.0.1                           eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Step 4 **show service-insertion type utd service-node-group**

Displays the status of service node groups.

Example:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Step 5 **show service-insertion type utd service-context**

Displays the AppNav and service node views.

Example:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

Step 6 **show utd engine standard config**

Displays the unified threat defense (UTD) configuration.

Example:

```
Device# show utd engine standard config


UTD Engine Standard Configuration:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug

Whitelist Signature IDs:
  28878
```

Step 7 **show utd engine standard status**

Displays the status of the utd engine.

Example:

```
Device# show utd engine standard status

Profile : High
System memory :
Usage : 8.00 %
Status : Green
Number of engines : 4
```

```
Engine Running CFT flows Health Reason
======================================================
Engine(#1): Yes 0 Green None
Engine(#2): Yes 0 Green None
Engine(#3): Yes 0 Green None
Engine(#4): Yes 0 Green None
======================================================

Overall system status: Green

Signature update status:
========================
Current signature package version: 2983.4.s
Last update status: Successful
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service not known'))
Next update scheduled at: None
Current status: Idle
```

Step 8 **show utd engine standard threat-inspection signature update status**

Displays the status of the signature update process.

Example:

```
Device# show utd engine standard threat-inspection signature update status

Current signature package version: 2983.4.s
Current signature package name: UTD-STD-SIGNATURE-2983-4-S.pkg
Previous signature package version: 29.0.c
----------------------------------------
Last update status: Successful
----------------------------------------
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last successful update method: Manual
Last successful update server: cisco
Last successful update speed: 3022328 bytes in 25 secs
----------------------------------------
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update method: Manual
Last failed update server: cisco
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service hnot known'))
----------------------------------------
Last attempted update time: Mon Aug  7 02:02:32 2017 UTC
Last attempted update method: Manual
Last attempted update server: cisco
----------------------------------------
Total num of updates successful: 1
Num of attempts successful: 1
Num of attempts failed: 3
Total num of attempts: 4
----------------------------------------
Next update scheduled at: None
----------------------------------------
Current status: Idle
```

Step 9 **show utd engine standard logging events**

Displays log events from the Snort sensor.

Example:

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Step 10 **clear utd engine standard logging events**

Example:

```
Device# clear utd engine standard logging events
```

Clears logged events from the Snort sensor.

Step 11 **show platform hardware qfp active feature utd config**

Displays information about the health of the service node.

Example:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 1 fo id 1 chunk id 8
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Step 12 **show platform software utd global**

Displays the interfaces on which UTD is enabled.

Example:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Enabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
All dataplane interfaces
```

Step 13 **show platform software utd interfaces**

Displays the information about all interfaces.

Example:

```
Device# show platform software utd interfaces

UTD interfaces
All dataplane interfaces
```

Step 14 **show platform hardware qfp active feature utd stats**

Displays dataplane UTD statistics.

Example:

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Pkts entered policy feature                        pkt                 228
                                                   byt               31083

Drop Statistics:

Service Node flagged flow for dropping                                  48
Service Node not healthy                                                62

General Statistics:

Non Diverted Pkts to/from divert interface                            32913
Inspection skipped - UTD policy not applicable                        48892
Policy already inspected                                               2226
Pkts Skipped - L2 adjacency glean                                         1
Pkts Skipped - For Us                                                    67
Pkts Skipped - New pkt from RP                                          102
Response Packet Seen                                                    891
Feature memory allocations                                              891
Feature memory free                                                     891
Feature Object Delete                                                   863

Service Node Statistics:
SN Health: Green
SN down                                                                  85
SN health green                                                          47
SN health red                                                            13

Diversion Statistics
redirect                                                              2226
encaps                                                                2226
decaps                                                                2298
reinject                                                              2250
decaps: Could not locate flow                                           72
Redirect failed, SN unhealthy                                           62
Service Node requested flow bypass drop                                 48
```

Step 15 **show utd engine standard statistics daq all**

Displays serviceplane data acquistion (DAQ) statistics.

Example:

```
Device# show utd engine standard statistics daq all
```

```
IOS-XE DAQ Counters(Engine #1):
--------------------------------
Frames received                         :0
Bytes received                          :0
RX frames released                      :0
Packets after vPath decap               :0
Bytes after vPath decap                 :0
Packets before vPath decap              :0
Bytes before vPath decap                :0
Frames transmitted                      :0
Bytes transmitted                       :0

Memory allocation                       :2
Memory free                             :0
Merged packet buffer allocation         :0
Merged packet buffer free               :0

VPL buffer allocation                   :0
VPL buffer free                         :0
VPL buffer expand                       :0
VPL buffer merge                        :0
VPL buffer split                        :0
VPL packet incomplete                   :0

VPL API error                           :0
CFT API error                           :0
Internal error                          :0
External error                          :0
Memory error                            :0
Timer error                             :0

Kernel frames received                  :0
Kernel frames dropped                   :0

FO cached via timer                     :0
Cached fo used                          :0
Cached fo freed                         :0
FO not found                            :0
CFT full packets                        :0


VPL Stats(Engine #1):
------------------------
```

Deploying Snort IPS Using Cisco Prime CLI Templates

You can use the Cisco Prime CLI templates to provision the Snort IPS deployment. The Cisco Prime CLI templates make provisioning Snort IPS deployment simple. To use the Cisco Prime CLI templates to provision the Snort IPS deployment, perform these steps:

Step 1 Download the Prime templates from the Software Download page, corresponding to the IOS XE version running on your system.

Step 2 Unzip the file, if it is a zipped version.

Step 3 From Prime, choose **Configuration** > **Templates** > **Features and Technologies**, select **CLI Templates**.

Step 4 Click **Import**.

Step 5 Select the folder where you want to import the templates to and click **Select Templates** and choose the templates that you just downloaded to import.

The following Snort IPS CLI templates are available:

- Copy OVA to Device—Use this template to copy the Snort IPS OVA file to the router file system.
- Delete OVA—Use this template to delete the copied Snort IPS OVA file from the router file system.
- Dynamic NAT—Use this template if Dynamic NAT (Network Address Translation) is configured in your environment and an Access List is used to select the NAT translation that needs to be modified for Snort IPS Management Interface IP.
- Dynamic NAT Cleanup—Use this template to delete the NAT configuration for Snort IPS.
- Dynamic PAT—Use this template if Dynamic PAT (Port Address Translation) is configured in your environment and an Access List is used to select the PAT translation that needs to be modified for Snort IPS Management Interface IP.
- Dynamic PAT Cleanup—Use this template to delete the PAT configuration for Snort IPS.
- IP Unnumbered—Use this template to configure Snort IPS and required Virtual-Service for IP Unnumbered deployment.
- IP Unnumbered Cleanup—Use this template to delete the configured Snort IPS Management interface with IP Unnumbered.
- Management Interface—Use this template if you would like to use System Management interface (e.g. GigabitEthernet0) to route Snort IPS Management traffic.
- Management Interface Cleanup—Use this template to delete the configured System Management interface (e.g. GigabitEthernet0) to route the Snort IPS Management traffic.
- Static NAT—Use this template to configure Snort IPS and required Virtual-Service for existing Static NAT deployment.
- Static NAT Cleanup—Use this template to delete the configured Snort IPS in a Static NAT deployment.
- Upgrade OVA—Use this template to upgrade Snort IPS OVA file.

Migrating to IOx Container

This section provides information about Cisco IOx and UTD migration to IOx for extending UTD support on Cisco 1000 Series Integrated Service Routers (ISRs). Cisco IOx combines Cisco IOS and the Linux OS for highly secure networking.

About Cisco IOx

Cisco IOx is an application platform that provides uniform and consistent hosting capabilities for various types of applications across various Cisco platforms. This platform brings together the networking operating

system-Cisco IOS, and the open source platform-Linux to bring together custom applications and interfaces on the network.

A virtual services container is a virtualized environment on a device. It is also referred to as a virtual machine (VM), virtual service, or container. You can install an application within a virtual services container. The application runs in the virtual services container of the operating system of a device. The application is delivered as an open virtual application (OVA), which is a tar file with a .ova extension. The OVA package is installed and enabled on a device through a command-line interface. Cisco Plug-in for OpenFlow is an example of an application that can be deployed within a virtual services container.

Virtual services container infrastructure that is used to host UTD OVA is not supported on Cisco 1100 Series ISRs. Currently, UTD supports both the containers. However, the OVA container feature support is continued on Cisco IOS XE Gibralter 16.10 release and is not supported for later releases.

Upgrading from Virtual Service Container to IOx

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. The Snort IPS is available as a virtual container service. You must download this OVA file on to the device and use the **virtual-service install** CLI to install the service.

For the UTD IOx infrastructure, the IOx based OVA is installed using IOx CLI commands. Before installing, start the IOx environment in global configuration mode.

The IOx based OVA is called a TAR file. You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

Perform the following steps to upgrade from virtual service to IOx container:

Step 1 **no activate**

Example:

```
Device# configure terminal
Device (config)# virtual-service utd
Device (config-virt-serv)# no activate
Device (config-virt-serv)# exit
Device (config)# no virtual-service utd
```

Deactivates virtual manager based virtual-service instance.

Step 2 **show virtual-service list**

Example:

```
Device# show virtual-service list
```

Displays the status of all applications installed on the virtual service container. Ensure that virtual service instance is deactivated.

Step 3 **virtual-service uninstall name** *virtual-service instance*

Example:

```
Device# virtual-service uninstall name utd
```

Uninstall virtual manager based virtual-service instance. Ensure that virtual service instance does not show up when you run **show virtual-service list** command.

Step 4 **iox**

Example:

```
Device# configure terminal
Device (config)# iox
Device (config)# end
```

Starts the IOx environment in Global Configuration mode.

Step 5 **app-hosting install appid** *name* **package** *bootflash:<tarfile>*

Example:

```
Device# app-hosting install appid UTD package bootflash:utd.tar
Device#
```

Copies and installs Iox based OVA tar file on to the device.

Step 6 **show app-hosting list**

Example:

```
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      DEPLOYED
Device#
```

Displays the status of the installation. Ensure that the application is deployed.

Step 7 **app-hosting activate appid** *name*

Example:

```
Device# app-hosting activate appid UTD
```

Activates the IOx based TAR file on the device.

Step 8 **show app-hosting list**

Example:

```
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      ACTIVATED

Device#
```

Displays the status of the activation. Ensure that the application is activated.

Step 9 **app-hosting start appid** *name*

Example:

```
Device# app-hosting start appid UTD
Device# show app-hosting list | in UTD
```

Starts the IOx based OVA.

Step 10 **show app-hosting list**

Example:

```
Example:
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      RUNNING
```

```
Device#
```

Displays the status of the start. Ensure that the application is running.

Example of IOx Configuration

Following is the example configuration of IOx:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# iox
Device(config)# interface VirtualPortGroup0
Device(config-if)# no shutdown
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup1
Device(config-if)# no shutdown
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# exit
Device(config)# app-hosting appid utd
Device(config-app-hosting)# app-vnic gateway0 virtualportgroup 0 guest-interface 0
Device(config-app-hosting-gateway0)# guest-ipaddress 192.0.2.2 netmask 255.255.255.252
Device(config-app-hosting-gateway0)# exit
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 1 guest-interface 1
Device(config-app-hosting-gateway1)# guest-ipaddress 192.0.2.6 netmask 255.255.255.252
Device(config-app-hosting-gateway1)# exit
Device(config-app-hosting)# app-resource package-profile custom
Device(config-app-hosting)# start
Device(config-app-hosting)# exit
Device(config)# exit
Device#
```

Troubleshooting Snort IPS

Traffic is not Diverted

Problem Traffic is not diverted.

Possible Cause Vitual-service may not be activated.

Solution Check whether the virtual-service is activated by using the **show virtual-service list** command. The following is sample output from the command:

```
Device# show virtual-service list

Virtual Service List:


Name Status Package Name
------------------------------------------------------------------------------
snort Activated utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Possible Cause Unified threat defense (UTD) may not be enabled for specified interface or interfaces.

Solution Use the **show platform software utd global** command to verify if UTD is enabled for the interface:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Disabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
GigabitEthernet0/0/0
```

Possible Cause The service node may not be working properly.

Solution Use the **show platform hardware qfp active feature utd config** command to verify if the health of the service node is green:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 0 fo id 0 chunk id 4
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Possible Cause The Snort process may not be activated.

Solution Use the **show virtual-service detail** command to verify if the Snort process is up and running:

```
Device# show virtual-service detail

Virtual service UTDIPS detail
  State                  : Activated
  Owner                  : IOSd
  Package information
    Name                 : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path                 : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name               : UTD-Snort-Feature
      Installed version  : 1.0.1_SV2982_XE_16_3
      Description        : Unified Threat Defense
    Signing
      Key type           : Cisco development key
      Method             : SHA-1
    Licensing
      Name               : Not Available
      Version            : Not Available

  Detailed guest status

----------------------------------------------------------------------
Process               Status            Uptime           # of restarts
----------------------------------------------------------------------
```

```
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk                : 736 MB
    Memory              : 1024 MB
    CPU                 : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                          Next Hop                              Intf.
-------------------------------------------------------------------------------------
0.0.0.0/0                             48.0.0.1                             eth2
0.0.0.0/0                             47.0.0.1                             eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Possible Cause The AppNav tunnel may not be activated.

Solution Use the **show service-insertion type utd service-node-group** and **show service-insertion type utd service-context** commands to verify if the AppNav tunnel is activated.

Solution The following is sample output from the **show service-insertion type utd service-node-group** command:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Solution The following is sample output from the **show service-insertion type utd service-context** command:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

Possible Cause Check data plane UTD statistics for the status of the traffic. If the traffic is not diverted, the number of packets diverted and rejected will be zero. If the numbers are nonzero, then traffic diversion is happening, and the Snort sensor is resending packets back to the dataplane.

Solution Use the **show platform hardware qfp active feature utd stats** commands to verify the status of the traffic.

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Active Connections                                                  29
TCP Connections Created                                         712910
UDP Connections Created                                             80
Pkts entered policy feature                         pkt           3537977
                                                    byt         273232057
Pkts entered divert feature                         pkt           3229148
                                                    byt         249344841
Pkts slow path                                      pkt            712990
                                                    byt          45391747
Pkts Diverted                                       pkt           3224752
                                                    byt         249103697
Pkts Re-injected                                    pkt           3224746
                                                    byt         249103373
….
```

Signature Update is not Working

Problem Signature update from Cisco Borderless Software Distribution (BSD) server is not working.

Possible Cause Signature update may have failed due to various reasons. Check for the reason for the last failure to update the signatures.

Solution Use the **show utd engine standard threat-inspection signature update status** command to display the reason for the last failure to update the signatures:

```
Device# show utd eng standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
----------------------------------------
Last update status: Failed
----------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
----------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
----------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
----------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
----------------------------------------
Next update scheduled at: None
----------------------------------------
Current status: Idle
```

Possible Cause Domain Name System (DNS) is not configured correctly.

Solution Use the **show running-config | i name-server** command to display the name server details:

```
Device#  show run | i name-server

ip name-server 10.104.49.223
```

Possible Cause System error—Failed to process the username and password combination.

Solution Ensure that you have provided the correct credentials for signature package download.

Signature Update from the Local Server is not Working

Problem Signature update from the local server not working.

Possible Cause Last failure Reason: Invalid scheme—only HTTP/HTTPS supported.

Solution Ensure that you have provided the HTTP or secure HTTP (HTTPS) as the local download method.

Possible Cause Last failure Reason: Name or service not known.

Solution Ensure that the hostname or IP address provided for the local server is correct.

Possible Cause Last failure Reason: Credentials not supplied.

Solution Ensure that you have provided the credentials for local HTTP/HTTPS server.

Possible Cause Last failure Reason: File not found.

Solution Ensure that the signature file name or URL that you have provided is correct.

Possible Cause Last failure Reason: Download corrupted.

Solution

- Verify whether the retry signature update is corrupted as the previous signature download.
- Ensure that the correct signature package is available.

Logging to IOSd Syslog is not Working

Problem Logging to IOSd syslog is not working.

Possible Cause Logging to syslog may not be configured in the unified threat defense (UTD) configuration.

Solution Use the **show utd engine standard config** command to display the UTD configuration and to ensure that logging to syslog is configured.

```
Device# show utd engine standard config

UTD Engine Standard Configutation:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
```

```
  Level     : debug

Whitelist Signature IDs:
  28878
```

Solution Use the following **show utd engine standard logging events** command to display the event logs for the UTD engine.

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Logging to an External Server is not Working

Problem Logging to an external server is not working.

Possible Cause Syslog may not be running on the external server.

Solution Verify whether syslog server is running on the external server. Configure the following command on the external server to view its status:

```
ps -eaf | grep syslog

 root 2073 1 0 Apr12 ? 00:00:02 syslogd -r -m
```

Possible Cause Connectivity between unified threat defense (UTD) Linux Container (LXC) and external server may be lost.

Solution Verify the connectivity from the management interface to the external syslog server.

UTD Conditional Debugging

Conditional debugging is supported by multi-tenancy for Unified Threat Defense. For further details about how to configure conditional debugging, see:

http://www.cisco.com/c/en/us/td/docs/routers/asr1000/troubleshooting/guide/Tblshooting-xe-3s-asr-1000-book.html#task_AC969BB06B414DCBBDEF7ADD29EF8131

Additional References for Snort IPS

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Snort IPS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 219: Feature Information for Snort IPS

Feature Name	Releases	Feature Information
Snort IPS	Cisco IOS XE 3.16.1S, 3.17S and later releases	The Snort IPS feature, enables Intrusion Prevention System (IPS) and Intrusion Detection System (IDS) for branch offices on Cisco IOS XE-based platforms. This feature uses the open source Snort solution to enable IPS and IDS.
VRF support on Snort IPS	Cisco IOS XE Denali 16.3.1	Supports Virtual Fragmentation Reassembly (VFR) on Snort IPS configuration.
Snort IPS support on Cisco Cloud Services Router 1000v Series	Cisco IOS XE Denali 16.3.1	Cisco Cloud Services Router 1000v Series supports Snot IPS.
UTD Snort IPS Enhancements for 16.4 Release	Cisco IOS XE Everest 16.4.1	The UTD Snort IPS enhancements for 16.4 release adds a feature for displaying the list of active signatures.
Threat Inspection Alerts Visibility UTD Serviceability enhancements	Cisco IOS XE Fuji 16.8.1	This feature provides summary of threat inspection alerts. The following commands are introduced: • **show utd engine standard logging statistics threat-inspection** • **show utd engine standard logging statistics threat-inspection** *detail* Following commands are modified as part of UTD Serviceability Enahancement: • **show utd engine standard status** • **show utd engine standard threat-inspection signature update status**
UTD (IPS and URL filtering) migration to IOX Containers	Cisco IOS XE Gibraltar 16.10.1	UTD is supported on Cisco 1100 Series ISRs by migrating virtual service container to IOx from OVA.

CHAPTER 165

Web Filtering

The Web Filtering feature enables the user to provide controlled access to Internet websites or Interanet sites by configuring the domain-based or URL-based policies and filters on the device. The user can configure the web filtering profiles to manage the web access. The Web Filtering feature is implemented using the container service and it is similar to the Snort IPS solution.

Web Filtering can either allow or deny access to a specific domain or URL based on:

- Allowed list and Blocked list—These are static rules, which helps the user to either allow or deny domains or URLs. If the same pattern is configured under both the allowed list and blocked list, the traffic will be allowed.
- Category—URLs can be classified into multiple categories such as News, Social Media, Education, Adult and so on. Based on the requirements, user has the option to block or allow one or more categories.
- Reputation—Each URL has a reputation score associated with it. The reputation score range is from 0-100, and it is categorized as: high-risk (reputation score (0-20), suspicious (0-40), moderate-risk (0-60), low-risk (0-80), and trustworthy (0-100). Based on the reputation score of a URL and the configuration, a URL is either blocked or allowed. If the user defines a reputation threshold through the CLI, all the URLs, with a reputation score lower than the user-defined threshold will be blocked.

Web Filtering

The Web Filtering feature enables the user to provide controlled access to Internet websites by configuring the domain-based or URL-based policies and filters on the device. Domain-based Filtering enables the user to control access to websites/servers at domain level, and URL-based Filtering enables the user to control access to websites at URL level. This section includes the following topics:

Domain-based Filtering

Domain-based filtering allows the user to control access to a domain by permitting or denying access based on the domain-based policies and filters configured on the device. When the client sends a DNS request through the Cisco Cloud Services Router 1000V Series, the DNS traffic is inspected based on the domain-based policies (allowed list/blocked list). Domains that are on the allowed list or blocked list will not be subjected to URL-based filtering even if they are configured. Graylist traffic does not match both allowed list and blocked list, and it is subjected to URL-based filtering if it is configured.

Domain-based Filtering Using Allowed List Filter

To allow the complete domain (cisco.com) without subjecting to any filtering, use the allowed list option . When a user makes a request to access a website using a browser, the browser makes a DNS request to get the IP address of the website. Domain filtering applies the filter on the DNS traffic. If the website's domain name matches to one of the allowed list patterns, domain filtering adds the website's address to the allowed list. The browser receives the IP address for the website and sends the HTTP(s) request to the IP address of the website. Domain filtering treats this traffic as allowed traffic. This allowed traffic is not further subjected to URL-based filtering even if it is configured. If the Snort IPS is configured, the traffic will be subjected to Snort IPS .

Domain-based Filtering Using Blocked List Filter

When a user want to block a complete domain (badsite.com), use the blocked list option. Domain filtering applies the filter on the DNS traffic. If the website's domain name matches to one of the patterns on the blocked list, domain filtering will send the configured blocked server's IP address in the DNS response to the end user instead of the actual resolved IP address of the website. The browser receives the blocked server's IP address as the IP address for the website and sends the HTTP(s) request to this IP address. This traffic is not further subjected to URL filtering or Snort IPS even if they are configured.The block server receives the HTTP(s) request and serves a block page to the end user. Also, when the DNS request matches a blocked list, all application traffic to that domain will be blocked.

Domain filtering is applied to all the DNS traffic even if the DNS requests are made in the context of non-HTTP(S) requests such as FTP, telnet, and so on. The blocked listed non-HTTP(S) traffic (FTP, telnet, and so on.) will also be forwarded to the block server. It is block server's responsibility to serve a block page or deny the request. You can configure an internal or external block server. For configuration steps, see Configure Domain-based Web Filtering with an External Block Server, on page 2120 and Configure Domain-based Web Filtering with a Local Block Server , on page 2121.

If the traffic is not part of the allowed list or on the blocked list during domain filtering, it will be subjected to URL filtering and Snort IPS if they are configured.

A user may consider using a combination of domain filtering allowed and blocked pattern lists to design the filters. For example, if a user wants to create an allowed list *www\.foo\.com*but also wants other domains on a blocked list, such as *www\.foo\.abc* and *www\.foo\.xyz*, configure the *www\.foo\.com* in the allowed list pattern and *www\.foo\.* in the blocked list pattern.

Note If you are using the *www* prefix in the allowed or blocked regex pattern, it can create a problem if the Server Name Indicator (SNI) returned in the client message doesn't match. For example, if you want to allow *www./foo./com* and SNI returns as *foo.com* only. We recommend not to include the *www* in the regex match.

URL-based Filtering

URL-based filtering allows a user to control access to Internet websites by permitting or denying access to specific websites based on the allowed list/blocked list, category, or reputation configuration. For example, when a client sends a HTTP/HTTP(s) request through the Cisco CSR 1000V Cloud Services Router, the HTTP/HTTP(s) traffic is inspected based on the URL filtering policies (Allowed list, Blocked list, Category, and Reputation). If the HTTP/HTTP(s) request matches the blocked list, the HTTP(s) request is blocked either by inline block page response or redirects the URL to a block server. If the HTTP/HTTP(s) request matches the allowed list, the traffic is allowed without further URL filtering inspection.

For HTTPS traffic, the inline block page will not be displayed. URL-based filtering will not decode any encoded URL before performing a lookup.

When there is no allowed list/blocked list configuration on the device, based on the category and reputation of the URL, traffic is allowed or blocked either using a block page or redirect URL for HTTP. For HTTP(s), there is no block page or redirect URL, the flow will be dropped.

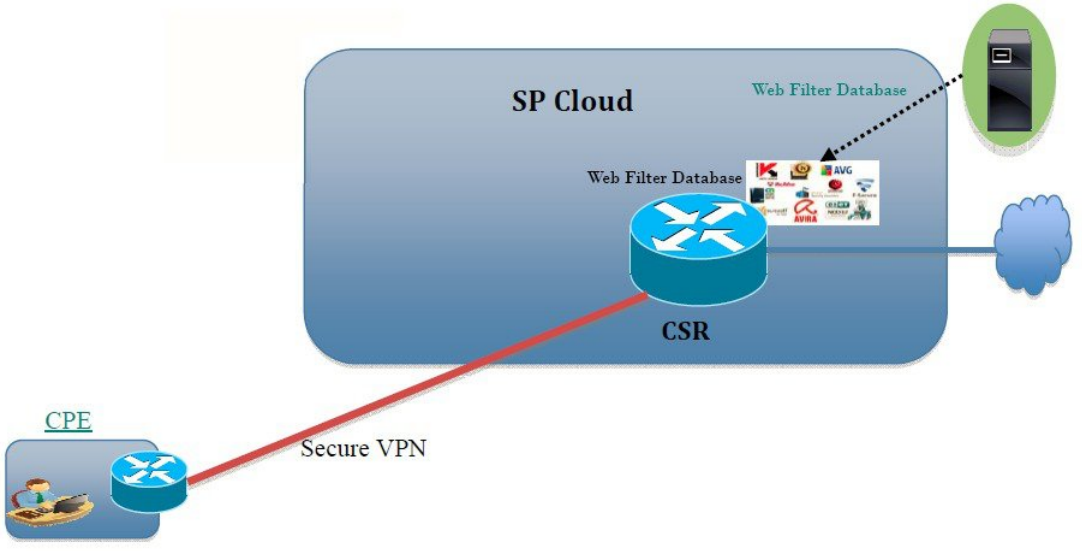
The URL database is downloaded from the cloud when the user configures the category/reputation-based URL filtering. The URL category/reputation database has only a few IP address based records and the category/reputation look up occurs only when the host portion of the URL has the domain name. After the full database is downloaded from the cloud, if there are any updates to the existing database, the incremental updates will be automatically downloaded in every 15 minutes. The complete database size is approximately 440 MB and the downloaded database should always synchronize with the cloud. The database will be invalid if the connection to the cloud is lost for more than 24 hours.

If the device does not get the database updates from the cloud, the fail-open option ensures that the traffic designated for URL filtering is not dropped. When you configure the fail-close option, all the traffic destined for URL filtering will be dropped when the cloud connectivity is lost.

Note The web filtering database is periodically updated from the cloud in every 15 minutes.

The figure illustrates the Web Filtering topology.

Figure 87: Web Filtering Network Topology



Virtual Service Resource Profiles for URL Filtering

The Cisco ISR 4000 Series Integrated Services Routers support *urlf-medium* and *urlf-high* resource profiles along with *urlf-low* profile. These profiles indicate the CPU and memory resources required to run the virtual service.

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	SP Memory	
CSR1000v, ISRv	*urlf-low*	25%	3 GB	8 GB (RAM)
	urlf-medium	50%	4 GB	8 GB (RAM)
	urlf-high	75%	6 GB	12 GB (RAM)

Cloud-Lookup

The Cloud-Lookup feature operates in single-tenancy mode to retrieve the category and reputation score of URLs that are not available in the local database. The Cloud-Lookup feature is enabled by default.

The Cloud-Lookup feature is an enhancement over the on-box database lookup feature. Earlier, the on-box database lookup feature allowed URLs that are not present in the on-box database and have a reputation score of 0. When Cloud-Lookup is enabled, the URLs that were allowed earlier may be dropped based on the reputation score and the configured block-threshold. In order to allow such URLs, one must add them to an allowed list. Category and reputation scores for different URLs from Cloud-Lookup are explained below.

There are two kinds of URLs:

- Name based URLs
- IP based URLs

When the Cloud-Lookup feature is enabled, the category and reputation score of unknown URLs are returned as follows:

Name based URLs

- Valid URL — corresponding category and reputation score is received.
- Unknown URL (new URL or unknown to the cloud) — category is 'uncategorized' and reputation score is 40
- Internal URLs with proper domain name (for example, internal.abc.com) — category and reputation score is based on the base domain name (abc.com from the example above).
- Completely internal URLs (for example, abc.xyz) — category is 'uncategorized' and reputation score is 40

IP based URLs

- Public hosted IP — corresponding category and reputation score is received.
- Private IP like 10.<>, 192.168.<> — category is 'uncategorized' and reputation score is 100
- Non-hosted/Non-routable IP — category is 'uncategorized' and reputation score is 40

The Cloud-Lookup score is different from the on-box database for these URLs (Unknown/Non-hosted/Non-routable/Internal URLs).

Note The Cloud-Lookup feature is not available in multi-tenancy mode.

Benefits of Web Filtering

The Web Filtering feature allows a user to provide controlled access to the internet by configuring domain and URL based policies and filters. It helps to secure the network by blocking malicious or unwanted websites.Web Filtering comprises of URL-based filtering and the Domain-based filtering. Domain-based filtering helps control access to websites/servers at domain level and the URL-based filtering helps control access to websites at URLs level. A user can use web filtering to add an individual URL to a blocked list or domain names and configure allowed listing policies for the same. A user can also provision to allow or block a URL based on reputation or category.

Prerequisites for Web Filtering

Before you configure the web filtering feature on the Cisco CSR 1000V Cloud Services Router, ensure that you have the following:

- The Cisco CSR 1000V Cloud Services Router runs the Cisco IOS XE Denali 16.3 software image or later.
- The Cisco CSR 1000V Cloud Services Router requires 2 vCPU, 8GB memory, and 2GB extra disk space for deploying the container service.

- The Cisco CSR 1000V Cloud Service Router must have a security K9 license to enable the web filtering feature.

Restrictions for Web Filtering

The following restrictions apply to the web filtering feature:

- This feature is only supported on Cisco CSR 1000V Cloud Services Router and it is not supported on Cisco 4000 Series Integrated Services Routers.
- The allowed list/blocked list pattern supports only regex pattern, and currently 64 patterns are supported for allowed list/blocked list. For more information on regex pattern, see the Regular Expressions chapter.
- Domain filtering supports only the IPv4 domains resolved through DNS protocol using IPv4 UDP transport. Domain filtering alerts are sent only to IOS syslog.
- Domain filtering with OpenDNS is not supported.
- URL filtering with Virtual Routing and Forwarding (VRF) is not supported.
- Domain filtering with CWS is not supported.
- Domain filtering does not support category and reputation.
- Local block server does not support serving HTTPS block page.When the URL filter tries to inject block page or redirect message, it does not support HTTPS traffic.
- When there is a username and password in the URL, URL filter does not remove them from the URL before matching the allowed list/blocked list pattern. However, the category/reputation lookup does not have this limitation and removes the username and password from the URL before lookup.
- HTTPS inspection is limited. Web filtering uses server certificate to obtain the URL/domain information. It is not possible to inspect the full URL path.
- UTD does not inter-operate with WCCP, and NBAR under inter-VRF scenario.
- Web filter profile names for URL, domain, block and sourcedb can have only alpha-numeric characters, dashes and underscores.
- If a virtual-service profile is modified, the virtual-service must be re-installed for the profile change to take effect.

How to Deploy Web Filtering

To deploy web filtering on supported devices, perform the following tasks:

Before you begin

- **Provision the device:** Identify the device to install the Web Filtering feature. This feature is supported on Cisco CSR 1000V Cloud Services Router.
- **Obtain the license:** The web filtering functionality is available only in security packages which require a security license to enable the service. Contact Cisco Support to obtain the license.

Step 1 Install and activate the virtual container service—How to Install and Activate the Virtual Container Service , on page 2119

Step 2 Configure the domain-based web filtering with an external block server—Configure Domain-based Web Filtering with an External Block Server, on page 2120

Step 3 Configure the domain-based web filtering with local block server—Configure Domain-based Web Filtering with a Local Block Server , on page 2121

Step 4 Configure the URL-based web filtering with a local block server—Configure URL-based Web Filtering with a Local Block Server, on page 2123

Step 5 Configure the URL-based web filtering with an Inline block server—Configure URL-based Web Filtering with an Inline Block Page, on page 2125

Step 6 Configure the Snort IPS/IDS—Configuring Domain/URL based Web Filtering and Snort IPS, on page 2126

How to Install and Activate the Virtual Container Service

To install and activate the virtual container service, perform the following task:

Step 1 Install the UTD OVA file—Installing the UTD OVA File, on page 2119.

Step 2 Configure the VirtualPortGroup interfaces and virtual-service—Configuring VirtualPortGroup Interfaces and Virtual Service, on page 2119.

Step 3 Activate the Snort virtual container service.

Installing the UTD OVA File

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. You must download this OVA file on to the router and use the virtual-service install CLI to install the service. The service OVA file is not bundled with the Cisco IOS XE Release images that are installed on the router. However, the OVA files may be preinstalled in the flash of the router.

You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

This is the sample configuration:

```
Device> enable
Device# virtual-service install name UTDIPS package harddisk:utd-ips-v102.ova media harddisk:

Device# show virtual-service list
Virtual Service List:
Name Status Package Name
------------------------------------------------------------------------------
snort Installed utdsnort.1_2_2_SV2982_XE_main.20160
```

Configuring VirtualPortGroup Interfaces and Virtual Service

You must configure two VirtualPortGroup interfaces and configure guest IP addresses for both interfaces.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

This is the sample configuration:

```
Device# configure terminal
evice(config)# interface VirtualPortGroup0
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup 1
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# exit
Device(config)# virtual-service UTDIPS

Device(config-virt-serv)# profile urlf-low (This is minimum requirement for web filtering
to work.)

Device(config-virt-serv)# vnic gateway VirtualPortGroup 0 (The IP-address configured in
VPG0 interface should have access to Internet over http(s).If the VPG0 interface does not
have access to Internet, the web filter database will not be updated.)
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup 1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.6
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate
Device(config-virt-serv)# end

Device# show virtual-service list
Virtual Service List:

Name                     Status              Package Name
------------------------------------------------------------------------------
snort                    Activated           utdsnort.1_2_2_SV2982_XE_main.20160
```

Configure Domain-based Web Filtering with an External Block Server

To configure domain-based web filtering with an external block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 2119.

Step 2 Configure the blocked list parameter-map:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern examplebook\.com
  pattern bitter\.com
```

Step 3 Configure the allowed list parameter-map:

```
parameter-map type regex domainfilter_whitelist_pmap1
  pattern example\.com
  pattern exmaplegogle\.com
```

Step 4 Configure the domain profile and associate the blocked list and allowed list parameter-maps:

```
utd web-filter domain profile 1
 blacklist
  parameter-map regex domainfilter_blacklist_pmap1
 whitelist
  parameter-map regex domainfilter_whitelist_pmap1
```

Step 5 (Optional) By default the domain filtering alerts are not enabled. Configure the alerts for the blocked list or allowed list, or both under the domain profile:

```
alert {all | blacklist | whitelist}
```

Step 6 Configure the external redirect-server under the domain profile:

```
redirect-server external x.x.x.x (This is the IP address that is used for serving block page when a
 page is on the blocked list)
```

Step 7 Configure the UTD engine standard with domain profile:

```
utd engine standard
 web-filter
  domain-profile 1
```

Step 8 Configure the UTD with engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configure domain-based web filtering with an external block server:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern examplebook\.com
  pattern bitter\.com
parameter-map type regex domainfilter_whitelist_pmap1
  pattern exmaplegogle\.com
  pattern exmaplegogle\.com
utd engine standard
  web-filter
    domain-profile 1
!
utd web-filter domain profile 1
  alert all
  blacklist
    parameter-map regex domainfilter_blacklist_pmap1
  whitelist
    parameter-map regex domainfilter_whitelist_pmap1
  redirect-server external 192.168.1.1
!
utd
  all-interfaces
  engine standard
```

Configure Domain-based Web Filtering with a Local Block Server

To configure domain-based web filtering with a local block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 2119.

Step 2 Configure a loopback interface or use any existing interface that the client can access:

```
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
exit
```

Step 3 Configure the UTD web filter with the local block server profile:

```
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
```

Step 4 Configure the blocked list parameter-map:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern bitter\.com
```

Step 5 Configure the allowed list parameter-map:

```
parameter-map type regex domainfilter_whitelist_pmap1
  pattern sweet\.com
```

Step 6 Configure the domain profile and associate the blocked list and allowed list parameter-maps:

```
utd web-filter domain profile1
 blacklist
  parameter-map regex domainfilter_blacklist_pmap1
 whitelist
  parameter-map regex domainfilter_whitelist_pmap1
```

Step 7 (Optional) By default the domain filtering alerts are not enabled. Configure the alerts for blocked list or allowed list, or both under the domain profile:

```
alert {all |blacklist | whitelist}
```

Step 8 Configure the redirect-server as local block server under the domain profile:

```
redirect-server local-block-server 1
```

Step 9 Configure the UTD engine standard with domain profile:

```
utd engine standard
 web-filter
  domain-profile 1
```

Step 10 Configure the UTD with engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configure a domain-based web filtering with a local block server:

```
interface loopback 110
  ip address 10.1.1.1 255.255.255.255
exit
parameter-map type regex domainfilter_blacklist_pmap1
  pattern bitter\.com
parameter-map type regex domainfilter_whitelist_pmap1
  pattern sweet\.com
utd engine standard
  web-filter
    domain-profile 1
!
```

```
utd web-filter block local-server profile 1
  block-page-interface Loopback110
  content text "Blocked by Web-Filter"
  http-ports 80
!
utd web-filter domain profile 1
  alert all
  blacklist
    parameter-map regex domainfilter_blacklist_pmap1
  whitelist
    parameter-map regex df_whitelist_pmap1
  redirect-server local-block-server 1
!
utd
  all-interfaces
  engine standard
```

Configure URL-based Web Filtering with a Local Block Server

To configure URL-based web filtering with a local block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 2119.

Step 2 Configure a loopback interface or use any existing interface that the client can access:

```
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
exit
```

Step 3 Configure the UTD web filter with the local block server profile:

```
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
```

Step 4 Configure the blocked list parameter-map:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern exmplee.com/sports
```

Step 5 Configure the allowed list parameter-map:

```
parameter-map type regex urlf_whitelist_pmap1
 pattern examplehoo.com/finance
```

Step 6 Configure the URL profile and do the following:

```
utd web-filter url profile 1
```

a) Associate the blocked list and allowed list parameter-maps:

```
  blacklist
    parameter-map regex urlf_blacklist_pmap1
  whitelist
    parameter-map regex urlf_whitelist_pmap1
```

b) Configure the alerts for blocked list, allowed list or both under the local block-server profile:

```
alert {all | blacklist | whitelist}
```

c) Configure the categories to be allowed or blocked:

```
categories allow
  sports
```

d) Configure the reputation block threshold:

```
reputation
  block-threshold high-risk
```

e) Configure the URL source database with the fail option:

```
sourcedb fail close
```

f) Configure the log level. The default option is error. When you set the option to **info** or **detail**, the performance may impact:

```
log level error
```

g) Configure local block server:block

```
block local-server 1
```

Step 7 Configure the UTD engine standard with URL profile:

```
utd engine standard
 web-filter
   url-profile 1
```

Step 8 Configure the UTD engine standard and enable the UTD on a global or specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configuration a URL-based web filtering with a local block server:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern examplee.com/sports
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
!
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
 exit
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
utd web-filter url profile 1
 blacklist
  parameter-map regex urlf_blacklist_pmap1
 whitelist
  parameter-map regex urlf_whitelist_pmap1
 alert all
 categories allow
  sports
 reputation
  block-threshold high-risk
```

```
 sourcedb fail close
 log level error
 block local-server 1
!
utd engine standard
 web-filter
  url-profile 1
!
utd
 all-interfaces
 engine standard
```

Configure URL-based Web Filtering with an Inline Block Page

To configure URL-based web filtering with an in-line block page, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 2119.

Step 2 Configure the blocked list parameter-map:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern exmaplegogle.com/sports
```

Step 3 Configure the allowed list parameter-map:

```
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
```

Step 4 Configure the UTD block page profile:

```
utd web-filter block page profile 1
 text "Blocked by Web-Filter URLF" (The other options are file and redirect-url)
```

Step 5 Configure the URL profile and do the following:

```
utd web-filter url profile 1
```

a) Associate the blocked list and allowed list parameter-maps:

```
blacklist
  parameter-map regex urlf_blacklist_pmap1
whitelist
  parameter-map regex urlf_whitelist_pmap1
```

b) Configure the alerts for blocked list, allowed list or both under the local block-server profile:

```
alert {all | blacklist | whitelist | categories-reputation}
```

c) Configure the categories to be allowed or blocked:

```
categories allow
  sports
```

d) Configure the reputation block threshold:

```
reputation
  block-threshold high-risk
```

e) Configure the URL source database with the fail option:

```
sourcedb fail close
```

f) Configure the log level. The default option is error. When you set the option to **info** or **detail**, the performance may impact:

```
log level error
```

g) Configure local block server:block

```
block local-server 1
```

Step 6 Configure the UTD engine standard with URL profile:

```
utd engine standard
 web-filter
   url-profile 1
```

Step 7 Configure the UTD engine standard and enable the UTD on a global or specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configuration an URL-based web filtering with an inline block server:

```
parameter-map type regex urlf_blacklist_pmap1
  pattern exmaplegogle.com/sports
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
!
utd web-filter block page profile 1
 text "Blocked by Web-Filter URLF"
!
utd web-filter url profile 1
 blacklist
  parameter-map regex urlf_blacklist_pmap1
 whitelist
  parameter-map regex urlf_whitelist_pmap1
 alert all
 categories allow
  sports
 reputation
  block-threshold high-risk
 sourcedb fail close
 log level error
!
utd engine standard
 web-filter
  url-profile 1
!
utd
 all-interfaces
 engine standard
```

Configuring Domain/URL based Web Filtering and Snort IPS

To configure Domain/URL based web filtering and Snort IPS, perform these steps:

Step 1 Configure the domain profile:

```
utd web-filter domain profile 1
```

Step 2 Configure the URL profile:

```
utd web-filter url profile 1
```

Step 3 Configure the threat-inspection under UTD engine standard:

```
utd engine standard
 threat-inspection
```

Step 4 Configure the web-filter under UTD engine standard with the domain and URL profiles:

```
utd engine standard
 logging syslog
 threat-inspection
  threat protection
  policy security
 signature update server cisco username xxx password QhLb]Z[ifMbFgLYgR]^KLDUZ
  signature update occur-at daily 0 0
  logging level error
 web-filter
  domain-profile 1
  url-profile 1
```

Step 5 Configure the UTD engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

Verifying the Web Filter Configuration

You can verify the Web Filtering configuration using the following commands:

```
Device# show utd engine standard config

UTD Engine Standard Configuration:
  Operation Mode : Intrusion Detection
  Policy         : Balanced

  Signature Update: Not Configured

  Logging:
    Server    :  IOS Syslog
    Level     : err (Default)
    Statistics    : Disabled

  Whitelist : Disabled
  Whitelist Signature IDs:

Web-Filter      : Enabled

  Whitelist :
    www.cisco.com
```

```
  Blacklist :
    www.hotstar.com

  Categories Action : Block
  Categories :
    Fashion and Beauty

  Block Profile:
   No config present

  Reputation Block Threshold : Moderate risk
  Alerts Enabled : Blacklist
  Cloud Lookup : Enabled
  Debug level : Error
Conditional debug level : Error
```

Troubleshooting Web Filtering

To collect the logs, use the **virtual-service move name "CONTAINER_NAME" log to bootflash:** command. You can troubleshoot issues that are related to enabling Web Filtering feature using the following commands on the device:

- **debug utd engine standard all**
- **debug utd engine standard climgr**
- **debug utd engine standard daq**
- **debug utd engine standard internal**
- **debug utd engine standard onep**
- **show utd engine standard logging events**

Note This tool will only show output for the configured URL filtering alerts/events. Users can configure the type of events and alerts they want to see in this output by following the steps in the section "Configuration Examples". For example, If you have configured "alert all", you will see "whitelist", "blacklist" and category & reputation events. If you configure only "alert whitelist", you will only see "whitelist" events."

For release 16.8.1, configuration error recovery on container is enhanced in order to apply configuration and signature updates to the container. With the improved error recovery, you can have:

- Greater robustness during configuration download to detect and act upon errors.
- Efficient way of handling signature and configuration updates occuring together.
- Early detect and recover from the loss of the oneP connection between IOSd and CLIMGR. For example, when CLIMGR crashes.
- Improved visibility to the detailed results of the (current or recent) configuration download, without requiring you to enable debugs.

The following site https://www.brightcloud.com/tools/url-ip-lookup.php can be used to validate how a website will be classified by our URL-Filtering feature.

Configuration Examples

The following example shows how to enable domain filtering on CSR 1000V Cloud Services Router:

```
Device# configure terminal
Device(config)# parameter-map type regex wlist1
Device(config-profile)# pattern google.com
Device(config-profile)# pattern cisco.com
Device(config-profile)# exit
Device(config)# parameter-map type regex blist1
Device(config-profile)# pattern exmaplehoo.com
Device(config-profile)# pattern bing.com
Device(config-profile)# exit
Device(config)# utd web-filter block local-server profile 1
Device(config--utd-webf-blk-srvr)# content file bootflash:test.utd.file
Device(config--utd-webf-blk-srvr)# end
```

For the local block server to work, HTTP server should be running. Use the ip http server command to configure the block server. The show ip http server status command displays the server status as enabled.

```
Device# show ip http server status
HTTP server status: Enabled
HTTP server port: 80
```

Example: Configuring Web Filter Domain Profile

The following example shows how to configure web filter domain profile:

```
Device(config)# utd web-filter domain profile 1
Device(config-utd-webfltr-domain)# blacklist
Device(config-utd-webf-dmn-bl)# parameter-map regex blist1
Device(config-utd-webf-dmn-bl)# whitelist
Device(config-utd-webf-dmn-wl)# parameter-map regex wlist1
Device(config-utd-webf-dmn-wl)# exit
Device(config-utd-webfltr-domain)# alert all
Device(config-utd-webfltr-domain)# redirect-server external 1.2.3.4
Device(config-utd-webfltr-domain)# exit
```

Configuring Web Filter URL Profile

The following example shows how to configure web filter URL profile:

```
Device(config)# utd web-filter url profile 1
Device(config-utd-webfltr-url)# blacklist
Device(config-utd-webf-url-bl)# parameter-map regex blist1
Device(config-utd-webf-url-bl)# whitelist
Device(config-utd-webf-url-wl)# parameter-map regex wlist1
Device(config-utd-webf-url-wl)# exit
Device(config-utd-webfltr-url)# categories allow
Device(config-utd-webf-url-cat)# news-and-media
Device(config-utd-webf-url-cat)# search-engines
Device(config-utd-webf-url-cat)# computer-and-internet-info
Device(config-utd-webf-url-cat)# computer-and-internet-security
Device(config-utd-webf-url-cat)# financial-services
Device(config-utd-webf-url-cat)# image-and-video-search
Device(config-utd-webf-url-cat)# job-search
Device(config-utd-webf-url-cat)#exit
Device(config-utd-webfltr-url)# alert all
```

```
Device(config-utd-webfltr-url)# reputation
Device(config-utd-webf-url-rep)# block-threshold suspicious
Device(config-utd-webf-url-rep)# exit
Device(config-utd-webfltr-url)# block local-server 1
Device(config-utd-webfltr-url)# exit
```

Configuring UTD Snort IPS/IDS Allowed List Signatures

The following example shows how to configure signature allowed lists:

```
Device(config)# utd threat-inspection whitelist
Device(config-utd-whitelist)# generator id 1 signature id 1
Device(config-utd-whitelist)# generator id 1 signature id 2
Device(config-utd-whitelist)# exit
```

Example: Configuring Web Filter Profile

The following example shows how to configure web filter profile:

```
Device(config)# utd engine standard
Device(config-utd-eng-std)# logging server 1.2.3.4
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)#threat protection
Device(config-utd-engstd-insp)# policy security
Device(config-utd-engstd-insp)# logging level emerg
Device(config-utd-engstd-insp)# whitelist
Device(config-utd-engstd-insp)# web-filter
Device(config-utd-engstd-webf)# domain-profile 1
Device(config-utd-engstd-webf)# url-profile 1
Device(config-utd-engstd-webf)# exit
```

Example: Alert Messages for Web Filtering Events

The following example shows alert messages for web filtering events:

```
016/06/02-14:44:41.061501 IST [**] [Instance_ID: 1] [**] Drop [**] UTD WebFilter Blacklist
 [**] [URL: www.edition.cnn.com/2016/03/31/asia/kolkata-bridge-collapse/index.html]
[Initiator_VRF: 0] {TCP} 1.0.0.9:56608 -> 2.0.0.29:80

2016/06/02-14:48:06.636270 IST [**] [Instance_ID: 1] [**] Pass [**] UTD WebFilter Whitelist
 [**] [URL: www.ndtv.com/index.html] [Initiator_VRF: 0] {TCP} 1.0.0.9:56611 -> 2.0.0.23:80

Jun 2 14:37:57.856 IST: %IOSXE-6-PLATFORM: F0: cpp_cp: QFP:0.0 Thread:000
TS:00000618422205723793 %UTD-6-UTD_DF_BLACKLIST_MATCH: UTD WebFilter Domain Blacklist [**]
 [Domain: www.cricinfo.com] [Matched Pattern: www.cricinfo.com] {UDP} 2.0.0.10:53 ->
1.0.0.9:55184

Jun 2 14:39:22.653 IST: %IOSXE-6-PLATFORM: F0: cpp_cp: QFP:0.0 Thread:000
TS:00000618507002407540 %UTD-6-UTD_DF_WHITELIST_MATCH: UTD WebFilter Domain Whitelist [**]
 [Domain: www.cricinfo.com] [Matched Pattern: www.cricinfo.com] {UDP} 2.0.0.10:53 ->
1.0.0.9:55286
```

Example: Unconfigure Cloud-Lookup

The following example shows how to unconfigure Cloud-Lookup feature in Web Filtering:

```
Device(config)# utd engine standard
Device(config-utd-eng-std)# web-filter
% Please ensure urlf-<low/medium/high> virtual-service profile is configured to use the
web-filter feature

Device(config-utd-engstd-webf)# no cloud-lookup
Device(config-utd-engstd-webf)# end
Device # exit
```

Additional References for Cisco Web Filtering

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
UCS E-Series Servers	http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/e/2-0/gs/guide/b_2_0_Getting_S

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Cisco Web Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 220: Feature Information for Cisco Web Filtering

Feature Name	Releases	Feature Information
Cisco Web Filtering	Cisco IOS XE Denali Release 16.3.1	The Web Filtering feature enables the user to provide controlled access to Internet websites by configuring the domain-based or URL-based policies and filters on the device. The user can configure the web filtering profiles to manage the web access.Web Filtering feature is implemented using the container service and it is similar to the Snort IPS solution.
UTD feature parity on ISRv UTD Serviceability Enhancements	Cisco IOS XE Fuji Release 16.8.1	Domain and URL filtering in both single-tenant and multi-tenant mode are supported for CSR. For ISRv, only single-tenant is supported. This feature is available on all models of the ENCS platforms. Error recovery feature in UTD is enhanced to allow the container to recover from internal error by initiating a bulk configuration download from IOS. The command **utd web-filter** *profile name* is modifed.
Web Root URL Filtering Enhancements	Cisco IOS XE Fuji Release 16.9.1	The URLF Virtual Resource Profiles in Web Filtering are supported only on platforms CSR1000v and ISRv. The URL Filtering supports cloud-lookup feature to search for the URLs in cloud that are not present in the database.

CHAPTER 166

Configuring Multi-Tenancy for Unified Threat Defense

Multi-tenancy for Unified Threat Defense provides Snort IPS and Web Filtering for multiple users. You can define policies for one or more tenants in a single Cisco CSR 1000v instance. Each policy can have a threat inspection profile and a web filtering profile. The following sections describe how to configure multi-tenancy for Unified Threat Defense. Many of the commands used in these configuration steps are similar to those used in configuring single-tenancy—see: Snort IPS, on page 2065 andWeb Filtering , on page 2113 .

Information About Multi-Tenancy for Unified Threat Defense

Multi-tenancy for Snort IPS and Web Filtering allows you to define policies for one or more tenants, in one Cisco CSR 1000v instance. This feature was introduced in Cisco IOS XE Everest 16.6.1.

Each tenant is a VPN routing and forwarding instance with one or more VPN routing and forwarding tables (VRFs). A Unified Threat Defense (UTD) policy is associated with a threat inspection profile and web filtering profile. Multiple tenants can share a UTD policy.

The system logs include the name of the VRF which allows you to produce statistics per-tenant.

The CLI commands used in multi-tenancy mode are similar to those used in single-tenancy mode (see Snort IPS, on page 2065 and Web Filtering , on page 2113). In multi-tenancy, you enter a sub-mode `utd engine standard multi-tenancy` and configure UTD policies, web filtering and threat-inspection profiles. After exiting the `utd engine standard multi-tenancy` sub-mode, the UTD policies are applied.

The benefits of web filtering and threat inspection (Snort IPS/IDS) are explained in the following sections:

- Benefits of Web Filtering , on page 2117
- Overview of Snort Virtual Service Interfaces, on page 2135

Web Filtering Overview

Web Filtering allows you to provide controlled access to the internet by configuring URL-based policies and filters. Web Filtering helps to control access to websites by blocking malicious or unwanted websites and therefore making the network more secure. You can blocked list individual URLs or domain names and configure allowed list policies for the same. You can also make provision to allow or block a URL based on reputation or category.

Snort IPS Overview

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the Snort engine to provide IPS and IDS functionalities.

Snort is an open source network IPS that performs real-time traffic analysis and generates alerts when threats are detected on IP networks. It can also perform protocol analysis, content searching or matching, and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, and so on. The Snort engine runs as a virtual container service on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series.

The Snort IPS feature works in the network intrusion detection and prevention mode that provides IPS or IDS functionalities. In the network intrusion detection and prevention mode, Snort performs the following actions:

- Monitors network traffic and analyzes against a defined rule set.
- Performs attack classification.
- Invokes actions against matched rules.

Based on your requirements, you can enable Snort either in IPS or IDS mode. In IDS mode, Snort inspects the traffic and reports alerts, but does not take any action to prevent attacks. In IPS mode, in addition to intrusion detection, actions are taken to prevent attacks.

The Snort IPS monitors the traffic and reports events to an external log server or the IOS syslog. Enabling logging to the IOS syslog may impact performance due to the potential volume of log messages. External third-party monitoring tools, which supports Snort logs, can be used for log collection and analysis.

Snort IPS Solution

The Snort IPS solution consists of the following entities:

- Snort sensor—Monitors the traffic to detect anomalies based on the configured security policies (that includes signatures, statistics, protocol analysis, and so on) and sends alert messages to the Alert/Reporting server. The Snort sensor is deployed as a virtual container service on the router.
- Signature store—Hosts the Cisco Signature packages that are updated periodically. These signature packages are downloaded to Snort sensors either periodically or on demand. Validated signature packages are posted to Cisco.com. Based on the configuration, signature packages can be downloaded from Cisco.com or a local server.

  The following domains are accessed by the router in the process of downloading the signature package from cisco.com:

  - api.cisco.com

- apx.cisco.com
- cloudsso.cisco.com
- cloudsso-test.cisco.com
- cloudsso-test3.cisco.com
- cloudsso-test4.cisco.com
- cloudsso-test5.cisco.com
- cloudsso-test6.cisco.com
- cloudsso.cisco.com
- download-ssc.cisco.com
- dl.cisco.com
- resolver1.opendns.com
- resolver2.opendns.com

Note If you are downloading signature packages from a local server to hold the signature packages, only HTTP is supported.

Signature packages must be manually downloaded from Cisco.com to the local server by using Cisco.com credentials before the Snort sensor can retrieve them.

The Snort container performs a domain-name lookup (on the DNS server(s) configured on the router) to resolve the location for automatic signature updates from Cisco.com or on the local server, if the URL is not specified as the IP address.

- Alert/Reporting server—Receives alert events from the Snort sensor. Alert events generated by the Snort sensor can either be sent to the IOS syslog or an external syslog server or to both IOS syslog and external syslog server. No external log servers are bundled with the Snort IPS solution.
- Management—Manages the Snort IPS solution. Management is configured using the IOS CLI. Snort Sensor cannot be accessed directly, and all configuration can only be done using the IOS CLI.

Overview of Snort Virtual Service Interfaces

The Snort sensor runs as a service on routers. Service containers use virtualization technology to provide a hosting environment on Cisco devices for applications.

You can enable Snort traffic inspection either on a per interface basis or globally on all supported interfaces. The traffic to be inspected is diverted to the Snort sensor and injected back. In Intrusion Detection System (IDS), identified threats are reported as log events and allowed. However, in Intrusion Prevention System (IPS), action is taken to prevent attacks along with log events.

The Snort sensor requires two VirtualPortGroup interfaces. The first VirtualPortGroup interface is used for management traffic and the second for data traffic between the forwarding plane and the Snort virtual container

service. Guest IP addresses must be configured for these VirtualPortGroup interfaces. The IP subnet assigned to the management VirtualPortGroup interface should be able to communicate with the Signature server and Alert/Reporting server.

The IP subnet of the second VirtualPortGroup interface must not be routable on the customer network because the traffic on this interface is internal to the router. Exposing the internal subnet to the outside world is a security risk. We recommend the use of 192.0.2.0/30 IP address range for the second VirtualPortGroup subnet. The use of 192.0.2.0/24 subnet is defined in RFC 3330.

You can assign the Snort virtual container service IP address on the same management network as the router on which the virtual service is running. This configuration helps if the syslog or update server is on the management network and is not accessible by any other interfaces

Restrictions for Configuring Multi-Tenancy for Unified Threat Defense

-
- Domain-based filtering is not supported.
- Up to 25 tenants are supported on each Cisco CSR 1000v instance.
- A maximum of 25 policies are supported.
- A maximum of 50,000 concurrent sessions are supported on a Cisco CSR 1000v.
-
- The blocked list/allowed list rules support only a regular expression (regex) pattern. Currently, 64 patterns are supported for each blocked list/allowed list rule. However, each tenant can have multiple rules.
- Local block server does not support serving HTTPS block page.When the URL filter tries to inject block page or redirect message, it does not support HTTPS traffic.
- When there is a username and password in the URL, URL filter does not remove them from the URL before matching the blocked list/allowed list pattern. However, the category/reputation lookup does not have this limitation and removes the username and password from the URL before lookup.
- HTTPS inspection is limited. Web filtering uses server certificate to obtain the URL/domain information. It is not possible to inspect the full URL path.
- UTD does not inter-operate with WCCP, and NBAR under inter-VRF scenario.
- The Snort IPS command `threat inspection profile` *`profile-name`* uses an alphanumeric profile-name, not an ID (number).

How to Configure Multi-Tenancy for Unified Threat Defense

To deploy multi-tenancy for Unified Threat Defense on supported devices, perform the following tasks:

Before you begin

Provision the device upon which you wish to install web filtering and threat inspection for multi-tenancy. This feature is currently only supported on the Cisco CSR 1000v.

Obtain the license. UTD is available only for routers running security packages and you will require a security license to enable the service. Contact Cisco Support to obtain a security license.

SUMMARY STEPS

1. Install and activate the virtual-service: Installing the UTD OVA File for Multi-Tenancy, on page 2137.
2. Configure the VirtualPortGroup interfaces and the virtual-service: How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy, on page 2138.
3. Configure the VRFs: How to Configure VRFs for Multi-Tenancy, on page 2141.
4. Configure threat inspection and web filtering for multi-tenancy: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 2142

DETAILED STEPS

Step 1 Install and activate the virtual-service: Installing the UTD OVA File for Multi-Tenancy, on page 2137.

Step 2 Configure the VirtualPortGroup interfaces and the virtual-service: How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy, on page 2138.

Step 3 Configure the VRFs: How to Configure VRFs for Multi-Tenancy, on page 2141.

Step 4 Configure threat inspection and web filtering for multi-tenancy: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 2142

Installing the UTD OVA File for Multi-Tenancy

The virtual-service OVA file is an Open Virtualization Archive file that contains a compressed, installable version of a virtual machine. You must download this OVA file to the router and then install the virtual-service. The virtual-service OVA file is not bundled with Cisco IOS XE release images that are installed on the router. OVA files may be available pre-installed in the router's flash memory.

For installing the OVA file, you must use a Cisco IOS XE image with a security license. During installation, the security license is checked.

Example of installing the virtual service:

```
Device> enable
Device# virtual-service install name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list

Name Status   Package Name
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

Example of upgrading the virtual service:

```
Device> enable
Device# virtual-service upgrade name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list
```

```
Name Status   Package Name
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

Example of uninstalling the virtual service:

```
Device> enable
Device# virtual-service uninstall name utd
Device# show virtual-service list

Virtual Service List:
```

How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy

As shown in this procedure, for multi-tenancy you must configure two VirtualPortGroup interfaces and guest IP addresses for both interfaces.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface VirtualPortGroup** *interface-number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface VirtualPortGroup** *interface-number*
7. **ip address** *ip-address mask*
8. **exit**
9. **virtual-service** *name*
10. **profile multi-tenancy**
11. **vnic gateway VirtualPortGroup** *interface-number*
12. **guest ip address** *ip-address*
13. **exit**
14. **vnic gateway VirtualPortGroup** *interface-number*
15. **guest ip address** *ip-address*
16. **exit**
17. **activate**
18. **end**
19. **show virtual-service list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface VirtualPortGroup** *interface-number* **Example:** `Device(config)# interface VirtualPortGroup 0`	Enters interface configuration mode and configures a VirtualPortGroup interface. This interface is used for management traffic when the management interface GigabitEthernet0 is not used.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.252`	Sets a primary IP address for an interface. This interface needs to be routable to the signature update server and external log server.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface VirtualPortGroup** *interface-number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. Configure a VirtualPortGroup interface. This interface is used for data traffic.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.1 255.255.255.252`	Sets a primary IP address for an interface. This IP address should not be routable to the outside network. The IP address is assigned from the recommended 192.0.2.0/30 subnet.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**virtual-service** *name* **Example:** `Device(config)# virtual-service utd`	Configures a virtual container service and enters virtual service configuration mode. The *name* argument is the logical name that is used to identify the virtual container service.
Step 10	**profile multi-tenancy** **Example:** `Device(config-virt-serv)#profile multi-tenancy`	Configures a resource profile. For multi-tenancy mode (Cisco CSR 1000v only), this `profile multi-tenancy` command must be configured.

	Command or Action	Purpose
Step 11	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	Enters the virtual-service virtual network interface card (vNIC) configuration mode. Creates a vNIC gateway interface for the virtual container service and maps the vNIC gateway interface to the virtual port group interface. This is the interface that was configured in Step 3.
Step 12	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 10.1.1.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 13	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 14	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 1`	Enters virtual-service vNIC configuration mode. Configures a vNIC gateway interface for the virtual container service and maps the interface to the virtual port group. The interface (*interface-number*) configured in Step 6) is used by the Snort engine for monitoring user traffic.
Step 15	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 192.0.2.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 16	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 17	**activate** **Example:** `Device(config-virt-serv)# activate`	Activates an application installed in a virtual container service.
Step 18	**end** **Example:** `Device(config-virt-serv)# end`	Exits virtual service configuration mode and returns to privileged EXEC mode.

Step 19	**show virtual-service list**
	Example:
	`Device# show virtual-service list`
	
	`Virtual Service List:`
	
	`Name  Status  Package Name`
	`----------`
	`utd  Activated  utdsnort.1.0.4_SV2983_XE_16_6.20170`	

How to Configure VRFs for Multi-Tenancy

This procedure describes the typical steps required for configuring VRFs for the tenants, which are later used in: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 2142.

Note For inter-VRF traffic, if the traffic flowing between two VRFs has ingress and egress interfaces configured for UTD, rules are applied to decide which VRF represents the session. The UTD policy for the selected VRF then applies to all packets in the inter-VRF traffic.

SUMMARY STEPS

1. **vrf definition** *vrf-name*
2. **rd** *route-distinguisher*
3. **address-family ipv4**
4. **exit address-family**
5. Repeat steps 1 to 4 for each VRF.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition 100`	Defines the name of the VRF and enters VRF configuration mode.
Step 2	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates the routing and forwarding tables and associates the *route-distinguisher* with the VRF instance named *vrf-name*. The router uses the route-distinguisher to identify the VRF to which a packet belongs. The route-distinguisher is of one of the following two types: • Autonomous System-related. An AS number xxx and an arbitrary number y—xxx:y • IP address-related. An IP address A.B.C.D and an arbitrary number y—A.B.C.D:y
Step 3	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions using the IP Version 4 address.
Step 4	**exit address-family** **Example:** `Device(config-vrf-af)# exit`	Exits address family configuration mode.
Step 5	Repeat steps 1 to 4 for each VRF.	

How to Configure Multi-Tenancy Web Filtering and Threat Inspection

To configure threat inspection (IPS/IDS) and web filtering for multi-tenancy (multiple tenants/VRFs), perform the following steps.

In this procedure, the definition of blocked list and allowed lists are shown in the initial steps 1 to 5. The main configuration steps (in UTD standard engine configuration mode for multi-tenancy) are shown in step 6 onwards.

Note For details about threat inspection and web filtering for single-tenancy, see Snort IPS, on page 2065 and Web Filtering , on page 2113.

Before you begin

Remove any existing single-tenancy UTD configuration, using the `no utd engine standard` command.

You must have previously configured a VRF for each tenant—see How to Configure VRFs for Multi-Tenancy, on page 2141.

Procedure

	Command or Action	Purpose
Step 1	**parameter-map type regex** *blacklist-name* **Example:** `Device(config)# parameter-map type regex urlf-blacklist1`	Defines a blocked list parameter map, which is used later in step 17.

Step 2	**pattern** *URL-name*
	Example:
	`Device(config-profile)# pattern www\.cnn\.com`
	`Device(config-profile)# pattern www\.msnbc\.com`
		Defines the URL to be on the blocked list. Note that the periods within *URL-name* must be preceded by an escape "\" character. Repeat this step to configure multiple URLs to be on the blocked list.

Step 3	**parameter-map type regex** *whitelist-name* **Example:** `Device(config-profile)# parameter-map type regex urlf-whitelist1`	Defines an allowed list parameter map, which is used later in step 20.
Step 4	**pattern** *URL-name* **Example:** `Device(config-profile)# pattern www\.nfl\.com`	Defines the URL(s) to be on the allowed list. Note that, for URLs on the blocked list, periods within *URL-name* must be preceded by an escape "\" character. Repeat this step to configure multiple URLs to be on the allowed list.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	

	Command or Action	Purpose
Step 6	**utd multi-tenancy** **Example:** `Device(config)# utd multi-tenancy`	This command acts a switch, in preparation for the following `utd engine standard multi-tenancy` command.
Step 7	**utd engine standard multi-tenancy** **Example:** `Device(config)# utd engine standard multi-tenancy`	Enters UTD standard engine configuration mode for multi-tenancy. **Note** Later. after you exit the UTD standard engine configuration mode in step 50, the policy configurations are applied.
Step 8	**web-filter sourcedb** *sourcedb-number* **Example:** `Device(config)# web-filter sourcedb 1`	Configures a web filtering sourcedb profile—*sourcedb-number*, which is numeric. This is used later in step 29.
Step 9	**logging level** {**alerts** | **critical** | **debugging** | **emergencies** | **errors** | **informational** | **notifications** | **warnings**} **Example:** `Device(config)# logging level errors`	Sets the level of system messages that are reported upon for web filtering events. Messages of the specified level and lower are reported. (Each level has a numeric value as shown in the table below.) ***Table 221: System Message Severity Levels*** **Level** / **Description** 0 – emergencies / System unusable 1 – alerts / Immediate action needed 2 – critical / Critical condition 3 – errors / Error condition 4 – warnings / Warning condition 5 – notifications / Normal but significant condition 6 – informational / Informational messages only 7 – debugging / Appears during debugging only
Step 10	**web-filter block local-server profile** *profile-id* **Example:** `Device(config-utd-multi-tenancy)# web-filter block local-server profile 1` The content text is displayed by the local server.	Configures the a local block server profile for web filtering. The range of values for *profile-id* is 1–255. See Configure URL-based Web Filtering with a Local Block Server. **Note** When configuring commands for multi-tenancy, compared to single-tenancy, you do not use the initial `utd` keyword.

	Command or Action	Purpose
Step 11	**block-page-interface loopback** *id* **Example:** `Device(config-utd-mt-webf-blk-srvr)# block-page-interface loopback 110`	Associates a loopback interface with this profile. The IP address of this loopback interface is then used as the IP address of the block local-server.
Step 12	**content text** *display-text* **Example:** `Device(config-utd-mt-webf-blk-srvr)# content text "Blocked by Web-Filter"`	Specifies the warning text that appears after a blocked page is accessed.
Step 13	**http-ports** *port-number* **Example:** `Device(config-utd-mt-webf-blk-srvr)# http-ports 80`	The http-ports value is a string of ports separated by commas. The nginx HTTP server listens to these ports.

Step 14	**web-filter block page profile** *profile-name*
	Example:
	`Device(config-utd-multi-tenancy)# web-filter block page profile 1`
	`Device(config-utd-mt-webf-block-urc)# text "this page is blocked"`
		See Configure URL-based Web Filtering with an Inline Block Page, on page 2125, except that the command used here for multi-tenancy does not use the `utd` keyword which is used for single-tenancy.).

Step 15	**web-filter url profile** *web-filter-profile-id*
	Example:
	`Device(config-utd-multi-tenancy)# web-filter url profile 1`
	`Device(config-utd-mt-webfltr-url)#`
		Specifies a URL profile for web filtering—*web-filter-profile-id*. Values: 1–255. After this command, you can configure alerts for blocked lists, allowed lists, and categories. For further information, see: Configure URL-based Web Filtering with an Inline Block Page.
		Note When configuring commands for multi-tenancy, compared to single-tenancy, you do not use an initial `utd` keyword.

Step 16	**blacklist** **Example:** `Device(config-utd-mt-webfltr-url)# blacklist`	Enters web filtering blocked list configuration mode.
Step 17	**parameter-map regex** *blacklist-name* **Example:** `Device(config-utd-mt-webf-url-bl)# parameter-map regex urlf-blacklist1`	Specifies a parameter-map regular expression using the blocked list that was defined earlier in step 1.

Step 18	**exit**
	Example:
	`Device(config-utd-mt-webf-url-bl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Exits web filtering blocked list configuration mode.

	Command or Action	Purpose

Step 19	**whitelist**
	Example:
	`Device(config-utd-mt-webfltr-url)# whitelist`
	`Device(config-utd-mt-webf-url-wl)#`
		Enters web filtering allowed list configuration mode.

Step 20	**parameter-map regex** *whitelist-name* **Example:** `Device(config-utd-mt-webf-url-wl)# parameter-map regex urlf-list1`	Specifies a parameter-map regular expression using the allowed list that was defined earlier in step 3.

Step 21	**exit**
	Example:
	`Device(config-utd-mt-webf-url-wl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Exits web filtering allowed list configuration mode.

Step 22	**exit**
	Example:
	`Device(config-utd-mt-webfltr-url)# exit`
	`Device(config-utd-multi-tenancy)#`
		Exits web filtering URL profile mode.

Step 23	**utd global** **Example:** `Device(config-utd-multi-tenancy)# utd global`	The commands entered for `utd global` apply to all tenants or policies e.g the commands shown below: `logging host syslog` and `threat inspection` for this Cisco CSR 1000v instance.

Step 24	**logging** {**host** *hostname* | **syslog**}
	Example:
	In this example, alerts are logged to a designated host log file.
	`Device(config-utd-mt-utd-global)# logging host systemlog1`
	Example:
	In this example, alerts are logged to IOS syslogs.
	`Device(config-utd-mt-utd-global)# logging syslog`
		The `logging` command specifies either a host name or IOS syslog, to which syslog messages are sent.

Step 25	**threat inspection** **Example:** `Device(config-utd-mt-utd-global)# threat inspection`	Enters global threat inspection mode.
Step 26	**signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-mt-utd-global-threat)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use www.cisco.com for signature updates, you must provide the username and password. If you use a local server for signature updates, based on the server settings you can provide the username and password.

	Command or Action	Purpose
		The router must be able to resolve the domain name by being connected to the internet.
Step 27	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-mt-utd-global-threat)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 28	**web-filter** **Example:** `Device(config-utd-mt-utd-global-threat)# web-filter`	This command, used in combination with the following `sourcedb` command, specifies the URL source database for web filtering.
Step 29	**sourcedb** *sourcedb-number* **Example:** `Device(config-utd-mt-utd-global-threat)# sourcedb 1`	Assigns a web filtering source database. Only one source database can be active.
Step 30	**exit** **Example:** `Device(config-utd-mt-utd-global-threat)# exit`	Exits threat inspection configuration mode.
Step 31	**exit** **Example:** `Device(config-utd-mt-global)# exit`	Exits global update configuration mode.
Step 32	**threat-inspection whitelist profile** *policy-name* **Example:** `Device(config-utd-multi-tenancy)# threat-inspection whitelist profile wh101`	Associates an allowed list profile with the policy currently being configured. A similar command is used in single-tenancy, but with a `utd` keyword.
Step 33	**signature id** *id* **Example:** `Device(config-utd-mt-list)# signature id 101`	Specify the ID *id* that you have previously identified as a threat; for example, after observing the ID in an alert log file. Repeat this command for multiple signature IDs.
Step 34	**exit** **Example:** `Device(config-utd-mt-whitelist)# exit`	Exits an allowed list configuration mode.
Step 35	**threat-inspection profile** *profile-name* **Example:** `Device(config-utd-multi-tenancy)# threat-inspection profile 101`	Configures a threat inspection profile, which can be reused by multiple tenants. You can configure multiple threat-inspection profiles. Within a profile you can configure multiple allowed lists. `profile-name` is alphanumeric.

	Command or Action	Purpose
Step 36	**threat** {**detection** | **protection** } **Example:** `Device(config-utd-mt-threat)# threat protection`	Specifies Intrusion Detection System (IDS) or Intrusion Prevention System (IPS) as the operating mode for the Snort engine. The default is **threat detection**
Step 37	**policy** {**balanced** | **connectivity** | **security**} **Example:** `Device(config-utd-mt-threat)# policy security`	Configures the security policy for the Snort engine. • The default security policy type is **balanced**.
Step 38	**logging level**{**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}	Provides logs in one of these categories: • alert—provides alert level logs (severity=2) • crit—critical level logs (severity=3) • debug—all logs (severity=8) • emerg—emergency level logs (severity=1) • err—error level logs (severity=4) Default. • info—info level logs (severity=7) • notice—notice level logs (severity=6) • warning—warning level logs (severity=5)
Step 39	**whitelist profile** *profile-name* **Example:** `Device(config-utd-mt-threat)# whitelist profile wh101`	You can also specify allowed list profiles in a profile only for allowed lists in another place—the `threat-inspection whitelist profile` command above. (Optional) Enables allowed lists under the UTD engine.
Step 40	**exit** **Example:** `Device(config-utd-mt-threat)# exit`	Exits threat inspection mode.
Step 41	Repeat steps 35 to 40 to add additional threat-inspection profiles.	
Step 42	**policy** *policy-name* **Example:** `Device(config-utd-multi-tenancy)# policy pol101`	Defines the policy that will be associated with multiple tenants. A threat detection (IPS) and web filtering profile are added to the policy.
Step 43	**vrf** [*vrf-name* | global] **Example:** This example shows the configuration of two tenants (VRFs) and two policies. `Device(config-utd-mt-policy)# vrf vrf101`	Repeat the `vrf vrf-name` command for each of the VRFs (tenants) that will use the UTD policy. These VRFs previously defined, see: How to Configure VRFs for Multi-Tenancy, on page 2141. Alternatively use `vrf global` to associate with the global (default) VRF and enables VRF under the interface.

	Command or Action	Purpose
Step 44	**all-interfaces** **Example:** `Device(config-utd-mt-policy)# all-interfaces`	(Optional) Associates all interfaces under the VRF with the policy.
Step 45	**threat-inspection profile** *profile-name* **Example:** `Device(config-utd-mt-policy)# threat-inspection profile 101`	(Optional) Associates the policy with a previously defined threat inspection profile, see Step 35.
Step 46	**web-filter url profile** *web-filter-profile-id* **Example:** `Device(config-utd-mt-policy)# web-filter url profile 1`	(Optional) Associates the policy with a previously defined web filtering profile, see step 15.
Step 47	**fail close** **Example:** `Device(config-utd-mt-policy)# fail close`	(Optional) Drops IPS/IDS packets on engine failure. Default is `fail open`.
Step 48	**exit**	Exits from policy configuration mode.
Step 49	Repeat steps 42 to 48 for each policy	
Step 50	**exit** **Example:** `Device(config-utd-multi-tenancy)# exit`	Exits the `utd engine standard multi-tenancy` mode. The policy configurations are applied, which may take a few minutes. During this time, further `utd engine standard multi-tenancy` configuration mode commands cannot be entered.

Step 51	**exit**
	Example:
	`Device(config)# exit`
	`Device#`	

Step 52	**show logging**
	Example:
	`Device(config)# show logging`
	`..UTD MT configuration download has started`
	`..UTD MT configuration download has completed`	

Step 53	**interface** *sub-interface* **Example:** `Device(config)# interface GigabitEthernet4.101`	Specify a sub-interface to be used for the tenant (VRF).
Step 54	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-if)# encapsulation dot1Q 101`	Applies a VLAN ID to the sub-interface.

	Command or Action	Purpose
Step 55	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrf101`	Associates a VRF instance with the sub-interface.
Step 56	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 111.0.0.1 255.255.255.0`	Specifies the sub-interface IP address of the VRF.
Step 57	**ip route** *ip-address subnet-mask sub-interface* **Example:** In this example, the VRF's subnet GigabitEthernet4.101 is linked to the global routing table using the static IP address 111.0.0.0 255.255.255.0. `Device(config-if)# ip route 111.0.0.0 255.255.255.0 GigabitEthernet4.101`	(Optional) This `ip route` command and the `ip route vrf` command in the following step are optional—you can use these steps if you want to configure route leaking using a static route between the VRF and the global routing table. This configures a static route to the VRF subnet from the VRF interface, so that the VRF subnet is accessible from the global routing table. For further information on configuring route leaking, see Route Leaking in MPLS/VPN Networks.
Step 58	**ip route vrf** *vrf-name ip-address subnet-mask* **global** **Example:** `Device(config-if)# ip route vrf vrf101 0.0.0.0 0.0.0.0 5.2.1.1 global`	(Optional) This step and the previous step are optional——you can use these steps if you want to configure route leaking using a static route between the VRF and the global routing table. For further information on configuring route leaking, see Route Leaking in MPLS/VPN Networks. Specifies the static VRF default route to the global routing table.
Step 59	**utd enable**	(Optional) Enables UTD on an interface. You can use this command if the `all-interfaces` command was not configured (in step 44).
Step 60	To configure a sub-interface for each tenant (VRF), repeat steps 53 to 59.	
Step 61	**exit**	Exits interface configuration mode.

The profiles for web filtering and threat inspection (IPS) have now been applied.

Example Configuration—Multi-Tenancy for Unified Threat Defense

This example shows a typical running configuration after configuring Multi-Tenancy for UTD for two tenants.

Note The following example mentions parameter maps `urlf-blacklist1` and `urlf-whitelist1`. The configuration of these parameter maps is not shown in the example. For further information on blocked list and approved list parameter-maps, see Configure URL-based Web Filtering with an Inline Block Page.

```
utd multi-tenancy
utd engine standard multi-tenancy
 web-filter block page profile 1
  text "This page is blocked"
 web-filter block page profile 2
  text "This page is blocked"
 web-filter url profile 1
  alert all
  blacklist
   parameter-map regex urlf-blacklist1
  whitelist
   parameter-map regex urlf-whitelist1
  categories block
   social-network
   sports
  block page-profile 1
  log level error
 web-filter url profile 2
  alert all
  blacklist
   parameter-map regex urlf-blacklist2
  categories block
   shopping
   news-and-media
   sports
   real-estate
   motor-vehicles
  block page-profile 2
  log level error
  reputation
   block-threshold low-risk
 web-filter sourcedb 1
  logging level error
 threat-inspection whitelist profile wh101
  signature id 101
 threat-inspection profile 101
  threat protection
  policy security
  logging level debug
  whitelist profile wh101
 threat-inspection profile 102
  threat detection
  policy security
  logging level debug
 utd global
  logging host 172.27.58.211
  logging host 172.27.58.212
  logging host 172.27.56.97
  threat-inspection
   signature update server cisco username abc password ]RDCe[B\^KFI_LgQgCFeBEKWP^SWZMZMb]KKAAB

   signature update occur-at daily 0 0
  web-filter
   sourcedb 1
 policy pol102
  vrf vrf102
  all-interfaces
  threat-inspection profile 102
  web-filter url profile 2
 policy pol101
  vrf vrf101
  all-interfaces
  threat-inspection profile 101
```

```
web-filter url profile 1
fail close
```

Verifying Unified Threat Defense Engine Standard Configuration

Use the following commands to verify your configuration.

SUMMARY STEPS

1. **enable**
2. **show utd multi-tenancy**
3. **show utd engine standard global**
4. **show utd engine standard status**
5. **show utd engine standard statistics**
6. **show utd engine standard statistics daq** [**dp** | **cp**]
7. **show utd engine standard statistics url-filtering** [**engine** | *no*]
8. **show utd engine standard statistics url-filtering vrf name** *vrf-name*
9. **show utd engine standard statistics internal**
10. **show utd engine standard logging event**
11. **show logging** | **include** CONFIG_DOWNLOAD
12. **show utd threat-inspection whitelist** [**profile** *profile-name*]
13. **show utd threat-inspection profile** *profile-name*
14. **show utd** [**policy** *profile-name*]
15. **show utd web-filter url** [**profile** *profile-name*]
16. **show utd web-filter block local-server** [**profile** *profile-name*]
17. **show utd web-filter sourcedb** [**profile** *profile-name*]
18. **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]
19. **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]
20. **show utd engine standard config web-filter url profile** *profile-name*
21. **show utd engine standard config** [**vrf name** *vrf-name*]
22. **show utd engine standard config threat-inspection profile** *profile-name*
23. **show utd engine standard threat-inspection signature update status**
24. **show platform software qfp active feature utd config** [**vrf**[{**id** *vrf-id* | **name** *vrf-name* | **global** }]
25. **show platform software utd interfaces**
26. **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]
27. **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]
28. **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]
29. **show platform hardware qfp active feature utd stats drop all**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device# enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show utd multi-tenancy**

Displays the current status of multi-tenancy.

Example:

```
Device# show utd multi-tenancy
Multitenancy is enabled
```

Step 3 **show utd engine standard global**

Displays the global settings for utd engine standard.

Example:

```
Device# show utd engine standard global
UTD Engine Standard Global: enabled
Threat-inspection: enabled
Web-filter: enabled
Logging:
```

Step 4 **show utd engine standard status**

Verify that the status of the UTD engine is Green.

Example:

```
Device# show utd eng standard status
Engine version       : 1.0.2_SV2983_XE_16_8

Profile              : Multi-tenancy
System memory        :
              Usage  : 3.50 %
              Status : Green
Number of engines    : 1

Engine        Running    CFT flows  Health     Reason
=========================================================
Engine(#1):   Yes        0          Green      None
=========================================================

Overall system status: Green

Signature update status:
========================
Current signature package version: 29.0.c
Last update status: Failed
Last successful update time: None
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update reason: [Errno 113] No route to host
Next update scheduled at: None
Current status: Idle
```

Step 5 **show utd engine standard statistics**

Example:

```
Device# show utd engine standard statistics
*************Engine #1*************
===============================================================================
Memory usage summary:
```

```
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
================================================================================
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
================================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)

<output removed for brevity>

Total: 49394
================================================================================
Action Stats:
Alerts: 65 ( 0.132%)
Logged: 65 ( 0.132%)
Passed: 0 ( 0.000%)
```

Step 6 **show utd engine standard statistics daq [dp | cp]**

Show Snort DAQ statistics.

Example:

```
Device# show utd engine standard statistics daq dp
IOS-XE DAQ Counters(Engine #1):
--------------------------------
Frames received 654101
Bytes received 549106120
RX frames released 654101
Packets after vPath decap 654101
Bytes after vPath decap 516510928
Packets before vPath encap 651686
Bytes before vPath encap 514800669
Frames transmitted 651686
Bytes transmitted 544447557

<output removed for brevity>
```

Example:

```
Device# show utd engine standard statistics daq cp
IOS-XE DAQ CP Counters(Engine #1):
-----------------------------------
Packets received :16353210
Bytes received :1112018252
Packets transmitted :16353210
Bytes transmitted :1700733776
Memory allocation :16353212
Memory free :16353210
CFT API error :0
VPL API error :0
Internal error :0
External error :0
Memory error :0
Timer error :0
```

```
RX ring full 0
CFT full 0
sPath lib flow handle exhausted 0
Memory status changed to yellow :1
Memory status changed to red :0
Process restart notifications :0
```

Step 7 **show utd engine standard statistics url-filtering** [**engine** | *no*]

Gives the URL statistics for all the tenants combined: the number of hits for sites on the blocked list, number of hits for sites on the allowed list, and the number of sites that are blocked by category block and reputation block.

Example:

```
Device# show utd engine standard statistics url-filtering
UTM Preprocessor Statistics
---------------------------
URL Filter Requests Sent:               377226166       379846771       381117940
URL Filter Response Received:           377009606       379622845       380892658
Blacklist Hit Count:                    0               0               0
Whitelist Hit Count:                    0               0               0

Reputation Lookup Count:                376859139       379458008       380706804
Reputation Action Block:                0               0               0
Reputation Action Pass:                 307             280             102
Reputation Action Default Pass:         376858832       379457728       380706702
Reputation Score None:                  376858832       379457728       380706702
Reputation Score Out of Range:          0               0               0

Category Lookup Count:                  376859139       379458008       380706804
Category Action Block:                  0               0               0
Category Action Pass:                   307             280             102
Category Action Default Pass:           376858832       379457728       380706702
Category None:                          376858832       379457728       380706702


Device# show utd engine standard statistics url-filtering engine1
UTM Preprocessor Statistics
-----------------------------------------------------
URL Filter Requests Sent:               377226166
URL Filter Response Received:           377009606
Blacklist Hit Count:                    0
Whitelist Hit Count:                    0

Reputation Lookup Count:                376859139
Reputation Action Block:                0
Reputation Action Pass:                 307
Reputation Action Default Pass:         376858832
Reputation Score None:                  376858832
Reputation Score Out of Range:          0

Category Lookup Count:                  376859139
Category Action Block:                  0
Category Action Pass:                   307
Category Action Default Pass:           376858832
Category None:                          376858832
```

Step 8 **show utd engine standard statistics url-filtering vrf name** *vrf-name*

Gives per-tenant URL statistics by using the additional parameters—**vrf name** *vrf-name* .

Example:

```
Device# show utd engine standard statistics url-filtering vrf name vrf101
UTM Preprocessor Statistics
---------------------------
URL Filter Requests Sent: 764
URL Filter Response Received: 764
Blacklist Hit Count: 3
Whitelist Hit Count: 44

Reputation Lookup Count: 764
Reputation Action Block: 0
Reputation Action Pass: 58
Reputation Action Default Pass: 706
Reputation Score None: 706
Reputation Score Out of Range: 0

Category Lookup Count: 764
Category Action Block: 5
Category Action Pass: 53
Category Action Default Pass: 706
Category None: 706
```

Step 9 **show utd engine standard statistics internal**

Example:

```
Device# show utd engine standard statistics internal
*************Engine #1*************
===============================================================================
Memory usage summary:
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
===============================================================================
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
===============================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)
VLAN: 49394 (100.000%)
IP4: 49394 (100.000%)
Frag: 0 ( 0.000%)
ICMP: 5 ( 0.010%)
UDP: 2195 ( 4.444%)
TCP: 47194 ( 95.546%)

<output removed for brevity>
```

Step 10 **show utd engine standard logging event**

Displays the logs which contains alerts and URLs that are either on the blocked or allowed list per VRF.

Example:

```
Device# show utd engine standard logging event

2017/08/04-16:01:49.205959 UTC [**] [Instance_ID: 1] [**] Drop [**]
```

```
UTD WebFilter Category/Reputation [**] [URL: www.cricinfo.com] ** [Category: Sports]
** [Reputation: 96] [VRF: vrf101] {TCP} 23.72.180.26:80 -> 111.0.0.254:53509
2017/08/04-16:02:12.253330 UTC [**] [Instance_ID: 1] [**] Pass [**]
 UTD WebFilter Whitelist [**] [URL: www.espn.go.com/m]
[VRF: vrf101] {TCP} 111.0.0.254:53511 -> 199.181.133.61:80
```

Step 11 **show logging** | **include** CONFIG_DOWNLOAD

Example:

```
show# logging | include CONFIG_DOWNLOAD
Aug 23 11:34:21.250 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has started
Aug 23 11:54:18.496 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has completed
```

Step 12 **show utd threat-inspection whitelist** [**profile** *profile-name*]

Displays all allowed list profiles or a specific allowed list profile.

Example:

```
Device# show utd threat-inspection whitelist
Whitelist Profile: wh101
Signature ID: 101
```

Example:

```
Device# show utd threat-inspection whitelist profile wh101
Whitelist Profile: wh101
Signature ID: 101
```

Step 13 **show utd threat-inspection profile** *profile-name*

Displays the details of a threat-inspection profile specified by the *profile-name*.

Example:

```
Device# show utd threat-inspection profile 101
Threat-inspection Profile: 101
Operational Mode: Intrusion Protection
Operational Policy: Security
Logging Level: debug
Whitelist Profile: wh101
```

Step 14 **show utd** [**policy** *profile-name*]

Displays all UTD policies or a specific UTD policy.

Example:

```
Device# show utd policy pol101
Policy name: pol101
VRF name: vrf101, VRF ID: 1
Global Inspection (across above VRFs): Enabled
Threat-inspection profile: 101
Web-filter URL profile: 1
Fail Policy: Fail-open
```

Step 15 **show utd web-filter url** [**profile** *profile-name*]

Displays all URL profiles or a specific profile.

Example:

```
Device# show utd web-filter url profile 1
URL Profile: 1
Alert: all
Blacklist Parameter Map Regex: urlf-blacklist1
Whitelist Parameter Map Regex: urlf-whitelist1
Block Categories:
dating
sports
Block Page Profile 1
Log level error
reputation block-threshold high-risk
```

Step 16 **show utd web-filter block local-server** [**profile** *profile-name*]

Displays all block page profiles or a specific block page profile.

Example:

```
Device# show utd web-filter block local-server profile 2
Block Local Server Profile: 2
Content text: "Blocked by Web-Filter"
HTTP ports: 80
```

Step 17 **show utd web-filter sourcedb** [**profile** *profile-name*]

Displays all sourcedb profiles or a specific sourcedb profile.

Example:

```
Device# show utd web-filter sourcedb
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0

SourceDB Profile: 2
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

Example:

```
Device# show utd web-filter sourcedb profile 1
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

Step 18 **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]

Displays serviceplane data acquistion (DAQ) statistics for all VRFs or a specific VRF.

Example:

The following example shows the serviceplane data acquisition statistics for VRF vrf101.

```
Device# show utd engine standard statistics daq dp vrf name vrf101
IOS-XE DAQ Counters(Engine #1):
--------------------------------
Frames received 374509
Bytes received 303136342
RX frames released 374509
Packets after vPath decap 374509
Bytes after vPath decap 284405526
Packets before vPath encap 372883
Bytes before vPath encap 283234522
Frames transmitted 372883
Bytes transmitted 300202270

Memory allocation 781856
Memory free 749636
Memory free via timer 29420
Merged packet buffer allocation 0
Merged packet buffer free 0

VPL buffer allocation 0
VPL buffer free 0
VPL buffer expand 0
VPL buffer merge 0
VPL buffer split 0
VPL packet incomplete 0

VPL API error 0
CFT API error 0
Internal error 52
External error 0
Memory error 0
Timer error 0

Kernel frames received 373590
Kernel frames dropped 0

FO cached via timer 0
Cached fo used 0
Cached fo freed 0
FO not found 0
CFT full packets 0
```

Step 19 **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]

Displays the details of a threat-inspection allowed list profile stored in a container.

Example:

```
Device# show utd engine standard config threat-inspection whitelist
UTD Engine Standard Configuration:

UTD threat-inspection whitelist profile table entries:
Whitelist profile: wh101
Entries: 1
```

Step 20 **show utd engine standard config web-filter url profile** *profile-name*

Displays the details of the web-filter profile stored in the container.

Example:

```
Device# show utd engine standard config web-filter url profile 1
UTD Engine Standard Configuration:
```

```
UTD web-filter profile table entries
Web-filter URL profile: 1
Whitelist:
www.espn.com
www.nbcsports.com
www.nfl.com
Blacklist:
www.cnn.com
Categories Action: Block
Categories:
Social Network
Sports
Block Profile: 1
Redirect URL: http://172.27.56.97/vrf101.html
Reputation Block Threshold: High risk
Alerts Enabled: Whitelist, Blacklist, Categories, Reputation
Debug level: Error
Conditional debug level: Error
```

Step 21 **show utd engine standard config** [**vrf name** *vrf-name*]

Displays the details of the UTD policy, threat-inspection profile and web-filter profile associated with a particular VRF.

Example:

```
Device# show utd engine standard config vrf name vrf101
UTD Engine Standard Configuration:

UTD VRF table entries:
VRF: vrf101 (1)
Policy: pol101
Threat Profile: 101
Webfilter Profile: 1
```

Step 22 **show utd engine standard config threat-inspection profile** *profile-name*

Displays the details of a specific threat-inspection profile.

Example:

```
Device# show utd engine standard config threat-inspection profile 101
UTD Engine Standard Configuration:

UTD threat-inspection profile table entries:
Threat profile: 101
Mode: Intrusion Prevention
Policy: Security
Logging level: Debug
Whitelist profile: wh101

Description:
Displays the details of a threat-inspection profile stored in the container.
```

Step 23 **show utd engine standard threat-inspection signature update status**

Shows the output of the current signature package version, previous signature package version, and last status update.

Example:

```
Device# show utd engine standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
```

```
---------------------------------------
Last update status: Failed
---------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
---------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
---------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
---------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Step 24 **show platform software qfp active feature utd config** [**vrf**[{**id** *vrf-id* | **name** *vrf-name* | **global** }]

Shows the service node statistics. The VRF information can only be shown in the case of multi-tenancy. Displays the data plane UTD configuration. In the following example the security context information is highlighted.

Example:

```
Device# Global configuration
  NAT64: disabled
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  Context Id: 0, Name: Base Security Ctx
   Ctx Flags: (0xf0000)
        Engine: Standard
        SN Redirect Mode : Fail-close, Divert
        Threat-inspection: Enabled, Mode: IPS
        Domain Filtering : Not Enabled
        URL Filtering    : Not Enabled
   SN Health: Green
```

Step 25 **show platform software utd interfaces**

Example:

```
Device# show platform software utd interfaces

UTD interfaces
All dataplane interfaces
```

Step 26 **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]

Show UTD datapath configuration and status.

Example:

```
Device# show platform hardware qfp active feature utd config vrf name vrf101
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 1 fo id 1 chunk id 8
  SN Health: Green
```

Step 27 **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]

Displays dataplane UTD statistics, including counts of zeros

clear—Clear Statistics

divert—Display AppNav Redirect Statistics

drop—Display Drop Statistics

general—Display General Statistics

summary—Display Summary Statistics

verbose—Display Verbose Statistics

vrf Display per VRF stats—The VRF information can only be entered if multi-tenancy is enabled.

id—display stats associated with the VRF id

name—display stats associated with the VRF with the provided name

global—display the stats associated with the global VRF (i.e vrf-id 0)

Example:

```
Device# show platform hardware qfp active feature utd stats

Summary Statistics:
TCP Connections Created 29893
UDP Connections Created 24402
ICMP Connections Created 796
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 715602
byt 562095214
Pkts entered divert feature pkt 662014
byt 516226302
Pkts slow path pkt 55091
byt 4347864
Pkts Diverted pkt 662014
byt 516226302
Pkts Re-injected pkt 659094
byt 514305557

Would-Drop Statistics:

Service Node flagged flow for dropping 258

General Statistics:
Non Diverted Pkts to/from divert interface 1022186
Inspection skipped - UTD policy not applicable 1081563
```

```
<output removed for brevity>
```

Example:

Step 28 **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]

Displays information about all VRFs or a specific VRF, taken from the summary option of the **show platform hardware qfp active feature utd stats** command.

Example:

```
Device# show platform hardware qfp active feature utd stats vrf name vrf101
Security Context: Id:1 Name: 1 : vrf101

Summary Statistics:
TCP Connections Created 18428
UDP Connections Created 13737
ICMP Connections Created 503
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 407148
byt 296496913
Pkts entered divert feature pkt 383176
byt 283158966
Pkts slow path pkt 32668
byt 2571632
Pkts Diverted pkt 383176
byt 283158966
Pkts Re-injected pkt 381016
byt 281761395


<output removed for brevity>
```

Step 29 **show platform hardware qfp active feature utd stats drop all**

Displays information from all the VRFs taken from the drop option of the **show platform** command.

Example:

```
Device# show platform hardware qfp active feature utd stats drop all

Would-Drop Statistics:

No diversion interface                                          0
No egress interface                                             0
Inspection service down                                         0
Could not find divert interface                                 0
Could not find divert fib                                       0
UTD FIB did not contain oce_chain                               0
Invalid IP version                                              0
IPS not supported                                               0
Re-inject Error                                                 0
Service Node flagged flow for dropping                       1225
Could not attach feature object                                 0
Could not allocate feature object                               0
Error getting feature object                                    0
Policy: could not create connection                             0
NAT64 Interface Look up Failed                                  0
Decaps: VPATH connection establishment error                    0
Decaps: VPATH could not find flow, no tuple                     0
Decaps: VPATH notification event error                          0
Decaps: Could not delete flow                                   0
Decaps: VPATH connection classification error                   0
```

```
Encaps: Error retrieving feature object                                  0
Encaps: Flow not classified                                              0
Encaps: VPATH connection specification error                             0
Encaps: VPATH First packet meta-data failed                              0
Encaps: VPATH No memory for meta-data                                    0
Encaps: VPATH Could not add TLV                                          0
Encaps: VPATH Could not fit TLV into memory                              0
Service Node Divert Failed                                               0
No feature object                                                        0
Service Node not healthy                                               123
Could not allocate VRF meta-data                                         0
Could not allocate debug meta-data                                       0
Packet was virtually fragmented (VFR)                                    0
IPv6 Fragment                                                            0
IPv4 Fragment                                                            0
```

Troubleshooting Multi-Tenancy for Unified Threat Defense

Traffic is not Diverted

Problem Traffic is not diverted.

Possible Cause Vitual-service may not be activated.

Solution Check whether the virtual-service is activated by using the **show virtual-service list** command. The following is sample output from the command:

```
Device# show virtual-service list

Virtual Service List:


Name Status Package Name
--------------------------------------------------------------------------------
snort Activated utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Possible Cause Unified threat defense (UTD) may not be enabled for specified interface or interfaces.

Solution Use the **show platform software utd global** command to verify if UTD is enabled for the interface:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Disabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
GigabitEthernet0/0/0
```

Possible Cause The service node may not be working properly.

Solution Use the **show platform hardware qfp active feature utd config** command to verify if the health of the service node is green:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 0 fo id 0 chunk id 4
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Solution Alternatively, in the case of multi-tenancy, you can use the **show platform hardware qfp active feature utd config vrf name** *vrf-name* command to verify if the health of the service node, for a specific VRF, is green:

```
Device# show platform hardware qfp active feature utd config vrf name vrf102
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  SN Health: Green
```

Possible Cause The Snort process may not be activated.

Solution Use the **show virtual-service detail** command to verify if the Snort process is up and running:

```
Device# show virtual-service detail

Virtual service UTDIPS detail
  State                : Activated
  Owner                : IOSd
  Package information
    Name               : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path               : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name             : UTD-Snort-Feature
      Installed version : 1.0.1_SV2982_XE_16_3
      Description      : Unified Threat Defense
    Signing
      Key type         : Cisco development key
      Method           : SHA-1
    Licensing
      Name             : Not Available
      Version          : Not Available

  Detailed guest status

------------------------------------------------------------------------
Process               Status            Uptime           # of restarts
------------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found
```

```
  Activated profile name: None
  Resource reservation
    Disk                : 736 MB
    Memory              : 1024 MB
    CPU                 : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                          Next Hop                            Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                             48.0.0.1                           eth2
0.0.0.0/0                             47.0.0.1                           eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Possible Cause The AppNav tunnel may not be activated.

Solution Use the **show service-insertion type utd service-node-group** and **show service-insertion type utd service-context** commands to verify if the AppNav tunnel is activated.

Solution The following is sample output from the **show service-insertion type utd service-node-group** command:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Solution The following is sample output from the **show service-insertion type utd service-context** command:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

Possible Cause Check data plane UTD statistics for the status of the traffic. If the traffic is not diverted, the number of packets diverted and rejected will be zero. If the numbers are nonzero, then traffic diversion is happening, and the Snort sensor is resending packets back to the dataplane.

Solution Use the **show platform hardware qfp active feature utd stats** command to verify the status of the traffic.

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Active Connections                                              29
TCP Connections Created                                     712910
UDP Connections Created                                         80
Pkts entered policy feature                      pkt       3537977
                                                 byt     273232057
```

```
Pkts entered divert feature                          pkt              3229148
                                                     byt            249344841
Pkts slow path                                       pkt               712990
                                                     byt             45391747
Pkts Diverted                                        pkt              3224752
                                                     byt            249103697
Pkts Re-injected                                     pkt              3224746
                                                     byt            249103373
….
```

Solution Alternatively, in the case of multi-tenancy, you can use the **show platform hardware qfp active feature utd stats vrf name** *vrf-name* command to verify the status of the traffic, for a specific VRF.

```
Device# show platform hardware qfp active feature utd stats vrf name vrf 101

Security Context:    Id:1    Name: 1 : vrf101

Summary Statistics:
Active Connections                                                      2
TCP Connections Created                                             34032
UDP Connections Created                                             11448
ICMP Connections Created                                               80
Pkts dropped                                         pkt              626
                                                     byt             323842
Pkts entered policy feature                          pkt              995312
                                                     byt            813163885
Pkts entered divert feature                          pkt              639349
                                                     byt            420083106
Pkts slow path                                       pkt               45560
                                                     byt             7103132
Pkts Diverted                                        pkt              638841
                                                     byt            419901335
Pkts Re-injected                                     pkt              630642
                                                     byt            412139098
….
```

Signature Update is not Working

Problem Signature update from Cisco Borderless Software Distribution (BSD) server is not working.

Possible Cause Signature update may have failed due to various reasons. Check for the reason for the last failure to update the signatures.

Solution Use the **show utd engine standard threat-inspection signature update status** command to display the reason for the last failure to update the signatures:

```
Device# show utd eng standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
----------------------------------------
Last update status: Failed
----------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
```

```
---------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
---------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
---------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Possible Cause Domain Name System (DNS) is not configured correctly.

Solution Use the **show running-config | i name-server** command to display the name server details:

```
Device#  show run | i name-server

ip name-server 10.104.49.223
```

Possible Cause System error—Failed to process the username and password combination.

Solution Ensure that you have provided the correct credentials for signature package download.

Signature Update from the Local Server is not Working

Problem Signature update from the local server not working.

Possible Cause Last failure Reason: Invalid scheme—only HTTP/HTTPS supported.

Solution Ensure that you have provided the HTTP or secure HTTP (HTTPS) as the local download method.

Possible Cause Last failure Reason: Name or service not known.

Solution Ensure that the hostname or IP address provided for the local server is correct.

Possible Cause Last failure Reason: Credentials not supplied.

Solution Ensure that you have provided the credentials for local HTTP/HTTPS server.

Possible Cause Last failure Reason: File not found.

Solution Ensure that the signature file name or URL that you have provided is correct.

Possible Cause Last failure Reason: Download corrupted.

Solution

- Verify whether the retry signature update is corrupted as the previous signature download.
- Ensure that the correct signature package is available.

Logging to IOSd Syslog is not Working

Problem Logging to IOSd syslog is not working.

Possible Cause Logging to syslog may not be configured in the unified threat defense (UTD) configuration.

Solution Use the **show utd engine standard config** command to display the UTD configuration and to ensure that logging to syslog is configured.

```
Device# show utd engine standard config

UTD Engine Standard Configutation:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug

Whitelist Signature IDs:
  28878
```

Solution Use the following **show utd engine standard logging events** command to display the event logs for the UTD engine.

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Logging to an External Server is not Working

Problem Logging to an external server is not working.

Possible Cause Syslog may not be running on the external server.

Solution Verify whether syslog server is running on the external server. Configure the following command on the external server to view its status:

```
ps -eaf | grep syslog

 root 2073 1 0 Apr12 ? 00:00:02 syslogd -r -m
```

Possible Cause Connectivity between unified threat defense (UTD) Linux Container (LXC) and external server may be lost.

Solution Verify the connectivity from the management interface to the external syslog server.

UTD Conditional Debugging

Conditional debugging is supported by multi-tenancy for Unified Threat Defense. For further details about how to configure conditional debugging, see:

http://www.cisco.com/c/en/us/td/docs/routers/asr1000/troubleshooting/guide/Tbls hooting-xe-3s-asr-1000-book.html#task_AC969BB06B414DCBBDEF7ADD29EF8131

PART XV

Umbrella

CHAPTER 167

Cisco Umbrella Integration

The Cisco Umbrella Integration feature enables cloud-based security service by inspecting the Domain Name System (DNS) query that is sent to the DNS server through the device. The security administrator configures policies on the Cisco Umbrella portal to either allow or deny traffic towards the fully qualified domain name (FQDN). Cisco device acts as a DNS forwarder on the network edge, transparently intercepts DNS traffic, and forwards the DNS queries to the Cisco Umbrella portal. This feature is available on Cisco IOS XE Denali 16.3 and later releases.

Restrictions for Cisco Umbrella Integration

- If an application or host uses IP address directly instead of DNS to query domain names, policy enforcement is not applied.
- When the client is connected to a web proxy, the DNS query does not pass through the Cisco device. In this case, the connector does not detect any DNS request and the connection to the web server bypasses any policy from the Cisco Umbrella portal.
- When the Cisco Umbrella Integration policy blocks a DNS query, the client is redirected to a Cisco Umbrella block page. HTTPS servers provide these block pages and the IP address range of these block pages is defined by the Cisco Umbrella portal.

- User authentication and identity is not supported in this release.
- The type A, AAAA, and TXT queries are the only records that are redirected. Other types of query bypasses the connector. Cisco Umbrella Connector maintains a list of IP address that is known for malicious traffic. When the Cisco Umbrella roaming client detects the destination of packets to those addresses, it forwards those addresses to Cisco Umbrella cloud for further inspection.
- Only the IPv4 address of the host is conveyed in the EDNS option.
- A maximum of 64 local domains can be configured, and the allowed domain name length is 100 characters.

Prerequisites for Cisco Umbrella Integration

Before you configure the Cisco Umbrella Integration feature, ensure that the following are met:

- The device has a security K9 license to enable Cisco Umbrella Integration.
- The device runs the Cisco IOS XE Denali 16.3 software image or later.
- Cisco Umbrella subscription license is available.
- The device is set as the default DNS server gateway and needs to ensure that the DNS traffic goes through the Cisco device.
- Communication for device registration to the Cisco Umbrella server is via HTTPS. This requires a root certificate to be installed on the router. To download this certificate directly from a link instead of pasting it in, you can find the certificate here: https://cacerts.digicert.com/DigiCertGlobalRootCA.crt.pem

Cloud-based Security Service Using Cisco Umbrella Integration

The Cisco Umbrella Integration feature provides cloud-based security service by inspecting the DNS query that is sent to the DNS server through the device. When a host initiates the traffic and sends a DNS query, the Cisco Umbrella Connector in the device intercepts and inspects the DNS query. If the DNS query is for a local domain, it forwards the query without changing the DNS packet to the DNS server in the enterprise network. If it is for an external domain, it adds an Extended DNS (EDNS) record to the query and sends it to Cisco Umbrella Resolver. An EDNS record includes the device identifier information, organization ID and client IP. Based on this information, Cisco Umbrella Cloud applies different policies to the DNS query.

Encrypting the DNS Packet

The DNS packet sent from the Cisco device to Cisco Umbrella Integration server must be encrypted if the EDNS information in the packet contains information such as user IDs, internal network IP addresses, and so on. When the DNS response is sent back from the DNS server, device decrypts the packet and forwards it to the host.

You can encrypt DNS packets only when the DNScrypt feature is enabled on the Cisco device.

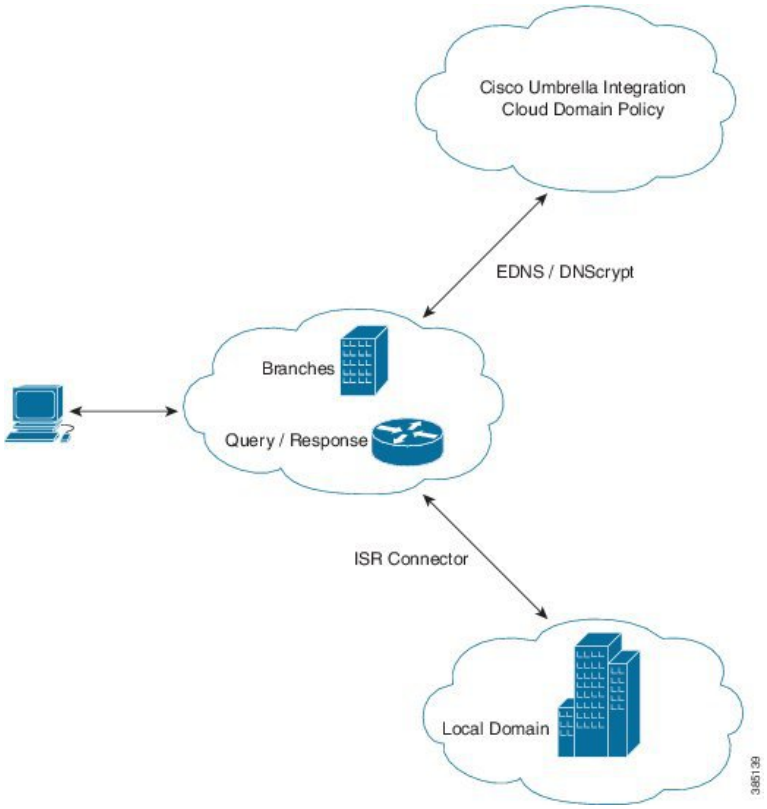
The Cisco device uses the following Anycast recursive Cisco Umbrella Integration servers:

- 208.67.222.222

- 208.67.220.220
- 2620:119:53::53
- 2620:119:35::35

The Figure 1 describes the Cisco Umbrella Integration topology.

Figure 88: Cisco Umbrella Integration Topology



Benefits of Cisco Umbrella Integration

Cisco Umbrella Integration provides security and policy enforcement at DNS level. It enables the administrator to split the DNS traffic and directly send some of the desired DNS traffic to a specific DNS server (DNS server located within the enterprise network). This helps the administrator to bypass the Cisco Umbrella Integration.

Configure the Cisco Umbrella Connector

To configure Cisco Umbrella Connector:

- Get the API token from the Cisco Umbrella registration server.

- Have the root certificate establish the HTTPS connection with the Cisco Umbrella registration server. Import the root certificate of DigiCert given below into the device using the **crypto pki trustpool import terminal** command.

```
-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR
TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr
hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls
YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=
-----END CERTIFICATE-----
```

- Verify that the PEM import is successful. A message is displayed after importing the certificate.

This is the sample configuration:

```
enable
configure terminal
parameter-map type umbrella global
 token AABBA59A0BDE1485C912AFE472952641001EEECC

exit
```

Registering the Cisco Umbrella Tag

To register the Cisco Umbrella tag, perform these steps:

1. Configure the umbrella parameter map as shown in the previous section.
2. Configure **umbrella out** on the WAN interface:

```
interface gigabitEthernet 0/0/1
 umbrella out
```

3. Configure **umbrella in** on the LAN interface:

```
interface gigabitEthernet 0/0/0.4
 umbrella in mydevice_tag
```

Note For the Cisco devices, the length of the hostname and umbrella tag should not exceed 49 characters.

4. After you configure **umbrella in** with a tag using the **umbrella in mydevice_tag** command, the device registers the tag to the Cisco Umbrella Integration portal.
5. The device initiates the registration process by resolving *api.opendns.com*. You need to have a name server (*ip name-server x.x.x.x*) and domain lookup (*ip domain-lookup*) configured on the device to successfully resolve the FQDN.

Note You should configure the **umbrella out** command before you configure **umbrella in** command. Registration is successful only when the port 443 is in *open* state and allows the traffic to pass through the existing firewall.

Configuring Cisco Device as a Pass-through Server

You can identify the traffic to be bypassed using domain names. In the Cisco device, you can define these domains in the form of regular expressions. If the DNS query that is intercepted by the device matches one of the configured regular expressions, then the query is bypassed to the specified DNS server without redirecting to the Cisco Umbrella cloud. This sample configuration shows how to define a regex parameter-map with a desired domain name and regular expressions:

```
Device# configure terminal
Device(config)# parameter-map type regex dns_bypass
Device(config)# pattern www.fisco.com
Device(config)# pattern .*engineering.fisco.*

Attach the regex param-map with the openDNS global configuration as shown below:

Device(config)# parameter-map type umbrella global
Device(config-profile)# token AADDD5FF6E510B28921A20C9B98EEEFF
Device(config-profile)# local-domain dns_bypass
```

DNSCrypt, Resolver, and Public-key

- DNSCrypt
- Resover IP
- Public-Key

We recommend that you change the above parameters only when you perform certain tests in the lab. These parameters are reserved for future use. If you modify these parameters, it can affect the normal functioning of the device.

Resolver

The following commands change the redirection of DNS packets from the Cisco device to Cisco Umbrella cloud:

- **resolver ipv4** *1.1.1.1*
- **resolver ipv4** *1.1.1.2*

- **resolver ipv6** *1234::1*
- **resolver ipv6** *2345::1*

In this example, all the IPv4 DNS packets are redirected to 1.1.1.1 or 1.1.1.2 and IPv6 DNS packets are redirected to 1234::1 or 2345::1. You should remove the IP address to restore to the default values of the resolver. When you modify a resolver IP address, the following message is displayed:

```
User configured would overwrite defaults
Defaults are restored when no more user configured are present
```

With the default values of **208.67.222.222** and **208.67.220.220**, all DNS packets are redirected to Cisco Umbrella Anycast resolvers. The device uses the first default resolver IP address for all its redirection. When the Cisco device does not receive a response for three consecutive DNS queries, the device automatically switches to a different resolver IP address. This behavior remains the same for IPv6 resolver addresses.

Note IPv6 redirection is deferred and all IPV6 DNS packets are not redirected to Cisco Umbrella Anycast servers.

Public-key

Public-key is used to download the DNSCrypt certificate from Cisco Umbrella Integration cloud. This value is preconfigured to **B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79** which is the public-key of Cisco Umbrella Integration Anycast servers. If there is a change in the public-key and if you modify this command, then you have to remove the modified command to restore the default value. If you modify the value, the DNSCrypt certificate download may fail.

DNSCrypt

To disable DNSCrypt, use the **no dnscrypt** command and to re-enable DNSCrypt, use the **dnscrypt** command.

When the DNSCrypt is used, the DNS request packets size is more than 512 bytes. Ensure that these packets are allowed through the intermediary devices; otherwise, the response may not reach the intended recipients.

Verifying the Cisco Umbrella Connector Configuration

Verify the Cisco Umbrella Connector configuration using the following commands:

```
Router# show umbrella config
Umbrella Configuration
==================
   Token:  AAC1A2555C11B2B798FFF3AF27C2FB8F001CB7B2
   OrganizationID: 1882034
   Local Domain Regex parameter-map name: NONE
   DNSCrypt: Enabled
  Public-key: B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79

   UDP Timeout: 5 seconds
   Resolver address:
       1. 208.67.220.220
       2. 208.67.222.222
       3. 2620:119:53::53
       4. 2620:119:35::35
```

```
   Umbrella Interface Config:
       Number of interfaces with "opendns out" config: 1
         1. GigabitEthernet0/0/0
             Mode     :  OUT
             VRF      : global(Id: 0)
       Number of interfaces with "opendns in" config: 1
         1. GigabitEthernet0/0/1
             Mode      : IN
             Tag       : test
             Device-id : 010a6aef0b443f0f
             VRF       : global(Id: 0)

Device# show umbrella deviceid
Device registration details
Interface Name          Tag       Status   Device-id
GigabitEthernet0/0/1     guest  200 SUCCESS 010a7ba73bd216d1


Device#show umbrella dnscrypt
    DNSCrypt: Enabled
Public-key: B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79
Certificate Update Status:
Last Successful Attempt : 10:55:40 UTC Apr 14 2016
Last Failed Attempt : 10:55:10 UTC Apr 14 2016
Certificate Details:
Certificate Magic : DNSC
Major Version : 0x0001
Minor Version : 0x0000
Query Magic : 0x717744506545635A
Serial Number : 1435874751
Start Time : 1435874751 (22:05:51 UTC Jul 2 2015)
End Time : 1467410751 (22:05:51 UTC Jul 1 2016)
Server Public Key :
ABA1:F000:D394:8045:672D:73E0:EAE6:F181:19D0:2A62:3791:EFAD:B04E:40B7:B6F9:C40B
Client Secret Key Hash :
BBC3:409F:5CB5:C3F3:06BD:A385:78DA:4CED:62BC:3985:1C41:BCCE:1342:DF13:B71E:F4CF
Client Public key :
ECE2:8295:2157:6797:6BE2:C563:A5A9:C5FC:C20D:ADAF:EB3C:A1A2:C09A:40AD:CAEA:FF76
NM key Hash :
F9C2:2C2C:330A:1972:D484:4DD8:8E5C:71FF:6775:53A7:0344:5484:B78D:01B1:B938:E884
```

Troubleshooting Cisco Umbrella Integration

Troubleshoot issues that are related to enabling Cisco Umbrella Integration feature using these commands:

- **debug umbrella device-registration**
- **debug umbrella config**
- **debug umbrella dnscrypt**

Depending on the OS, run either of these two commands from the client device:

- The **nslookup -type=txt debug.umbrella.com** command from the command prompt of the Windows machine
- The **nslookup -type=txt debug.umbrella.com** command from the terminal window or shell of the Linux machine

```
nslookup -type=txt debug.opendns.com 8.8.8.8
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
debug.opendns.com text = "server r6.mum1"
debug.opendns.com text = "device 010A826AAABB6C3D"
debug.opendns.com text = "organization id 1892929"
debug.opendns.com text = "remoteip 171.168.1.7"
debug.opendns.com text = "flags 436 0 6040 39FF000000000000000"
debug.opendns.com text = "originid 119211936"
debug.opendns.com text = "orgid 1892929"
debug.opendns.com text = "orgflags 3"
debug.opendns.com text = "actype 0"
debug.opendns.com text = "bundle 365396"
debug.opendns.com text = "source 72.163.220.18:36914"
debug.opendns.com text = "dnscrypt enabled (713156774457306E)"
```

Configuration Examples

This example shows how to enable Cisco Umbrella Integration:

Deploying Cisco Umbrella Integration Using Cisco Prime CLI Templates

You can use the Cisco Prime CLI templates to provision the Cisco Umbrella Integration deployment. The Cisco Prime CLI templates make provisioning Cisco Umbrella Integration deployment simple.

Note The Cisco Prime CLI templates is supported only on Cisco Prime version 3.1 or later.

To use the Cisco Prime CLI templates to provision the Cisco Umbrella Integration deployment, perform these steps:

Step 1 Download the Cisco Prime templates corresponding to the Cisco IOS XE version running on your system.

Step 2 Unzip the file, if it is a zipped version.

Step 3 From Cisco Prime Web UI, choose **Configuration** > **Templates** > **Features and Technologies**, and then select **CLI Templates** (User Defined).

Step 4 Click **Import**.

Step 5 Select the folder where you want to import the templates and click Select Templates and choose the templates that you just downloaded.

Step 6 The following Cisco Umbrella Integration templates are available:

- Umbrella—Use this template to provision Umbrella Connector on the device.
- Umbrella Cleanup—Use this template to remove previously configured Umbrella Connector.

Additional References for Cisco Umbrella Integration

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Cisco Umbrella Integration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 222: Feature Information for Cisco Umbrella Integration

Feature Name	Releases	Feature Information
Cisco Umbrella Integration	Cisco IOS XE Everest Release 16.6.1	The Cisco Umbrella Integration feature enables cloud-based security service by inspecting the DNS query that is sent to any DNS server through Cisco devices. The security administrator configures policies on the Umbrella cloud to either allow or deny traffic towards the fully qualified domain name (FQDN). This feature is supported only on Cisco ISRs.

PART XVI

User Security

CHAPTER 168

Cisco IOS Login Enhancements-Login Block

The Cisco IOS Login Enhancements (Login Block) feature allows users to enhance the security of a router by configuring options to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected.

The login block and login delay options introduced by this feature can be configured for Telnet or SSH virtual connections. By enabling this feature, you can slow down “dictionary attacks” by enforcing a “quiet period” if multiple failed connection attempts are detected, thereby protecting the routing device from a type of denial-of-service attack.

Note Whenever you want to use the AAA "quiet-mode" feature, you have to configure the aaa new-model using the **aaa new-model** command.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Cisco IOS Login Enhancements

Protecting Against Denial of Service and Dictionary Login Attacks

Connecting to a routing device for the purposes of administering (managing) the device, at either the User or Executive level, is most frequently performed using Telnet or SSH (secure shell) from a remote console (such as a PC). SSH provides a more secure connection option because communication traffic between the user's device and the managed device are encrypted. The Login Block capability, when enabled, applies to both Telnet connections and SSH connections.

The automated activation and logging of the Login Block and Quiet Period capabilities introduced by this feature are designed to further enhance the security of your devices by specifically addressing two well known methods that individuals use to attempt to disrupt or compromise networked devices.

If the connection address of a device is discovered and is reachable, a malicious user may attempt to interfere with the normal operations of the device by flooding it with connection requests. This type of attack is referred to as an attempted Denial-of-Service, because it is possible that the device may become too busy trying to process the repeated login connection attempts to properly handle normal routing services or will not be able to provide the normal login service to legitimate system administrators.

The primary intention of a dictionary attack, unlike a typical DoS attack, is to actually gain administrative access to the device. A dictionary attack is an automated process to attempt to login by attempting thousands, or even millions, of username/password combinations. (This type of attack is called a "dictionary attack" because it typically uses, as a start, every word found in a typical dictionary as a possible password.) As scripts or programs are used to attempt this access, the profile for such attempts is typically the same as for DoS attempts; multiple login attempts in a short period of time.

By enabling a detection profile, the routing device can be configured to react to repeated failed login attempts by refusing further connection request (login blocking). This block can be configured for a period of time, called a "quiet period". Legitimate connection attempts can still be permitted during a quiet period by configuring an access-list (ACL) with the addresses that you know to be associated with system administrators.

Login Enhancements Functionality Overview

Delays Between Successive Login Attempts

A Cisco device can accept virtual connections as fast as they can be processed. Introducing a delay between login attempts helps to protect the Cisco device against malicious login connections such as dictionary attacks and DoS attacks. Delays can be enabled in one of the following ways:

- Via the **auto secure** command. If you enable the AutoSecure feature, the default login delay time of one second is automatically enforced.
- Via the **login block-for** command. You must enter this command before issuing the **login delay** command. If you enter only the **login block-for** command, the default login delay time of one second is automatically enforced.
- Via the new global configuration mode command, **login delay**, which allows you to specify a the login delay time to be enforced, in seconds.

Login Shutdown If DoS Attacks Are Suspected

If the configured number of connection attempts fail within a specified time period, the Cisco device will not accept any additional connections for a "quiet period." (Hosts that are permitted by a predefined access-control list [ACL] are excluded from the quiet period.)

The number of failed connection attempts that trigger the quiet period can be specified via the new global configuration mode command **login block-for**. The predefined ACL that is excluded from the quiet period can be specified via the new global configuration mode command **login quiet-mode access-class**.

This functionality is disabled by default, and it is not enabled if autosecure if enabled.

How to Configure Cisco IOS Login Enhancements

Configuring Login Parameters

Use this task to configure your Cisco device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

All login parameters are disabled by default. You must issue the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following defaults are enforced:

- A default login delay of one second
- All login attempts made via Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is issued.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **login block-for** *seconds* **attempts** *tries* **within** *seconds*
5. **login quiet-mode access-class** {*acl-name* | *acl-number*}
6. **login delay** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 4	**login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Router(config)# login block-for 100 attempts 2 within 100`	Configures your Cisco IOS XE device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.

Step 5	**login quiet-mode access-class** {*acl-name* | *acl-number*}
	Example:
	`Router(config)# login quiet-mode access-class myacl`
		(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the router when the router switches to quiet mode. When the router is in quiet mode, all login requests are deniedand the only available connection is through the console.
		If this command is not configured, then the default ACL **sl_def_acl** is created on the router. This ACL is hidden in the running configuration. Use the **show access-list sl_def_acl** to view the parameters for the default ACL.
		For example:
		`Router#show access-lists sl_def_acl`
		`Extended IP access list sl_def_acl`
		`10 deny tcp any any eq telnet`
		`20 deny tcp any any eq www`
		`30 deny tcp any any eq 22`
		`40 permit ip any any`

Step 6	**login delay** *seconds* **Example:** `Router(config)# login delay 10`	(Optional) Configures a delay between successive login attempts.

What to Do Next

After you have configured login parameters on your router, you may wish to verify the settings. To complete this task, see the following section "Verifying Login Parameters, on page 2189."

Verifying Login Parameters

Use this task to verify the applied login configuration and present login status on your router.

SUMMARY STEPS

1. **enable**
2. **show login failures**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show login failures** **Example:** `Router# show login`	Displays login parameters. • **failures** --Displays information related only to failed login attempts.

Examples

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Router# show login
No login delay has been applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps
Router NOT enabled to watch for login Attacks
```

The following sample output from the **show login** command verifies that the **login block-for**command is issued. In this example, the command is configured to block login hosts for 100 seconds if 16 or more login requests fail within 100 seconds; five login requests have already failed.

```
Router# show login
A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.
Router enabled to watch for login Attacks.
If more than 15 login failures occur in 100 seconds or less, logins will be disabled for
100 seconds.
Router presently in Watch-Mode, will remain in Watch-Mode for 95 seconds.
Present login failure count 5.
```

The following sample output from the **show login** command verifies that the router is in quiet mode. In this example, the **login block-for** command was configured to block login hosts for 100 seconds if 3 or more login requests fail within 100 seconds.

```
Router# show login
A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.
Router enabled to watch for login Attacks.
If more than 2 login failures occur in 100 seconds or less, logins will be disabled for 100
 seconds.
Router presently in Quiet-Mode, will remain in Quiet-Mode for 93 seconds.
Denying logins from all sources.
```

The following sample output from **show login failures** command shows all failed login attempts on the router:

```
Router# show login failures
Information about login failure's with the device
Username      Source IPAddr  lPort Count  TimeStamp
try1          10.1.1.1        23    1     21:52:49 UTC Sun Mar 9 2003
try2          10.1.1.2        23    1     21:52:52 UTC Sun Mar 9 2003
```

The following sample output from **show login failures** command verifies that no information is presently logged:

```
Router# show login failures
*** No logged failed login attempts with the device.***
```

Configuration Examples for Login Parameters

Setting Login Parameters Example

The following example shows how to configure your router to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests will be denied during the quiet period except hosts from the ACL “myacl.”

```
Router(config)# aaa new-model
Router(config)# login block-for 100 attempts 15 within 100
Router(config)# login quiet-mode access-class myacl
```

Additional References

Related Documents

Related Topic	Document Title
Configuring autosecure	AutoSecure feature module.
Security commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Secure Management/Administrative Access	Role-Based CLI Access feature module.

Standards

Standards	Title
None.	--

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco IOS Login Enhancements-Login Block

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 223: Feature Information for Cisco IOS Login Enhancements (Login Block)

Feature Name	Releases	Feature Configuration Information
Cisco IOS Login Enhancements	Cisco IOS XE Release 2.1	The Cisco IOS Login Enhancements (Login Block) feature allows users to enhance the security of a router by configuring options to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Service Aggregation Routers. The following commands were modified by this feature: **login block-for**, **login delay**, **login quiet-mode access-class**, **show login**.

CHAPTER 169

Configuring Security with Passwords, Privileges, and Logins

Cisco IOS based networking devices provide several features that can be used to implement basic security for CLI sessions using only the operating system running on the device. These features include the following:

- Different levels of authorization for CLI sessions to control access to commands that can modify the status of the networking device versus commands that are used to monitor the device
- Assigning passwords to CLI sessions
- Requiring users log in to a networking device with a username
- Changing the privilege levels of commands to create new authorization levels for CLI sessions

This module is a guide to implementing a baseline level of security for your networking devices. It focuses on the least complex options available for implementing a baseline level of security. If you have networking devices installed in your network with no security options configured, or you are about to install a networking device and you need help understanding the how to implement a baseline of security, this document will help you.

Restrictions for Configuring Security with Passwords, Privileges, and Logins

Your networking device must not be configured to use any local or remote authentication, authorization, and accounting (AAA) security features. This document describes only the non-AAA security features that can be configured locally on the networking device.

For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Securing User Services Configuration Guide Library*.

Restrictions and Guidelines for Reversible Password Types

- Password type 0 and type 7 are deprecated. So password type 0 and type 7, used for administrator login to Console, Telnet, SSH, webUI, and NETCONF, must be migrated to password type 8 or type 9.
- No action is required if username and password are type 0 and type 7 for local authentication such as CHAP, EAP and so on for ISG and Dot1x.
- Enable password type 0 and type 7 must be migrated to password type 8 or type 9.

Restrictions and Guidelines for Irreversible Password Types

- Password type 5 is deprecated. Password type 5 must be migrated to stronger password type 8 or type 9.
- For username secret password type 5 and for enable secret password type 5, migrate to type 8 or type 9.
- Secret password type 4 is not supported.

Information About Configuring Security with Passwords, Privileges, and Logins

Benefits of Creating a Security Scheme

The foundation of a good security scheme in the network is the protection of the user interfaces of the networking devices from unauthorized access. Protecting access to the user interfaces on your networking devices prevents unauthorized users from making configuration changes that can disrupt the stability of your network or compromise your network security.

The Cisco IOS XE features described in this document can be combined in many different ways to create a unique security scheme for each of your networking devices. Here are some possible examples that you can configure:

- You can enable non administrative users to run a subset of the administrative commands available on the networking device by lowering the entitlement level for the commands to the non administrative privilege level. This can be useful for the following scenarios:
  - ISPs that want their first-line technical support staff to perform tasks such as enabling new interfaces for new customers or resetting the connection for a customer whose connection has stopped passing traffic. See the Example: Configuring a Device to Allow Users to Shutdown and Enable Interfaces, on page 2229 section for an example of how to do this.
  - When you want your first-line technical support staff to have the ability to clear console port sessions that were disconnected improperly from a terminal server. See the Example: Configuring a Device to Allow Users to Clear Remote Sessions, on page 2226 section for an example of how to do this.

- When you want your first-line technical support staff to have the ability to view, but not change, the configuration of a networking device to facilitate troubleshooting a networking problem. See the Example: Configuring a Device to Allow Users to View the Running Configuration, on page 2227 section for an example of how to do this.

Cisco IOS XE CLI Modes

To aid in the configuration of Cisco devices, the Cisco IOS XE command-line interface is divided into different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on the mode you are in. Entering a question mark(**?**) at the system prompt (device prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The standard order in which a user would access the modes is as follows: user EXEC mode; privileged EXEC mode; global configuration mode; specific configuration modes; configuration submodes; and configuration subsubmodes.

Note The default configuration of a Cisco IOS XE software based networking device only allows you to configure passwords to protect access to user EXEC mode (for local, and remote CLI sessions) and privileged EXEC mode. This document describes how you can provide additional levels of security by protecting access to other modes, and commands, using a combination of usernames, passwords and the **privilege** command.

Most EXEC mode commands are one-time commands, such as **show** or **more** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. EXEC mode commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode* . In this mode, you can enter commands that configure general system characteristics. You also can use global configuration mode to enter specific configuration modes. Configuration modes, including global configuration mode, allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. For example, *interface configuration mode* , is a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of a given configuration mode. As an example, this chapter describes the *subinterface configuration mode* , a submode of the interface configuration mode.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your system (router, switch, or access server) does not find a valid system image to load when it is booting, the system will enter ROM monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup. ROMMON is not covered in this document because it does not have any security features available in it.

User EXEC Mode

When you start a session on a router, you generally begin in *user EXEC mode* , which is one of two access levels of the EXEC mode. For security purposes, only a limited subset of EXEC commands are available in

user EXEC mode. This level of access is reserved for tasks that do not change the configuration of the router, such as determining the router status.

If your device is configured to require users to log-in the log-in process will require a username and a password. You may try three times to enter a password before the connection attempt is refused.

User EXEC mode is set by default to privilege level 1. Privileged EXEC mode is set by default to privilege level 15. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 1 are a subset of those available at privilege level 15. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. You can move commands to any privilege level between 1 and 15 using the **privilege** command.

In general, the user EXEC commands allow you to connect to remote devices, change terminal line settings on a temporary basis, perform basic tests, and list system information.

To list the available user EXEC commands, use the following command:

Command	Purpose
`Device(config)#` **?**	Lists the user EXEC mode commands

The user EXEC mode prompt consists of the host name of the device followed by an angle bracket (>), as shown in the following example:

```
Device>
```

The default host name is generally Router, unless it has been changed during initial configuration using the **setup** EXEC command. You also change the host name using the **hostname** global configurationcommand.

Note Examples in Cisco IOS XE documentation assume the use of the default name of "Device." Different devices (for example, access servers) may use a different default name. If the device (router, access server, or switch) has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Device> ?

Exec commands:
 <1-99>           Session number to resume
 connect          Open a terminal connection
 disconnect       Disconnect an existing telnet session
 enable           Turn on privileged commands
 exit             Exit from Exec mode
 help             Description of the interactive help system
 lat              Open a lat connection
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from Exec mode and log out
 menu             Start a menu-based user interface
 mbranch          Trace multicast route for branch of tree
 mrbranch         Trace reverse multicast route to branch of tree
```

```
mtrace           Trace multicast route to group
name-connection  Name an existing telnet connection
pad              Open a X.29 PAD connection
ping             Send echo messages
resume           Resume an active telnet connection
show             Show running system information
systat           Display information about terminal lines
telnet           Open a telnet connection
terminal         Set terminal line parameters
tn3270           Open a tn3270 connection
trace            Trace route to destination
where            List active telnet connections
x3               Set X.3 parameters on PAD
```

The list of commands will vary depending on the software feature set and platform you are using.

Note You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, Cisco IOS XE documentation convention is to always present commands in lowercase.

Privileged EXEC Mode

In order to have access to all commands, you must enter *privileged EXEC mode* , which is the second level of access for the EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. In privileged EXEC mode, you can enter any EXEC command, because privileged EXEC mode is a superset of the user EXEC mode commands.

Because many privileged EXEC mode commands set operating parameters, privileged EXEC level access should be password protected to prevent unauthorized use. The privileged EXEC command set includes those commands contained in user EXEC mode. Privileged EXEC mode also provides access to configuration modes through the **configure**command, and includes advanced testing commands, such as **debug**.

Privileged EXEC mode is set by default to privilege level 15. User EXEC mode is set by default to privilege level 1. For more information see the User EXEC Mode, on page 2195. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 15 are a superset of those available at privilege level 1. You can move commands to any privilege level between 1 and 15 using the **privilege** command. See the Cisco IOS XE Privilege Levels, on page 2204 for more information on privilege levels and the **privilege** command.

The privileged EXEC mode prompt consists of the host name of the device followed by a pound sign(#), as shown in the following example:

```
Device#
```

To access privileged EXEC mode, use the following command:

Command	Purpose

`Device> enable`
 	
`Password`
 	
`Device# exit`
 	
`Device>`
	Enables privileged EXEC mode.
	• If a privileged EXEC mode password has been configured the system will prompt you for a password after you issue the enable command.
	• Use the exit command to leave privileged EXEC mode.

Note Privileged EXEC mode is sometimes referred to as "enable mode," because the **enable** command is used to enter the mode.

If a password has been configured on the system, you will be prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, privileged EXEC mode can be accessed only by a local CLI session (terminal connected to the console port).

If you attempt to access privileged EXEC mode on a router over a remote connection, such as a telnet connection, and you have not configured a password for privileged EXEC mode you will see the **% No password set** error message. For more information on remote connections see the Remote CLI Sessions, on page 2201. The system administrator uses the **enable secret** or **enable password**global configuration commands to set the password that restricts access to privileged EXEC mode. For information on configuring a password for privileged EXEC mode, see the Protecting Access to Privileged EXEC Mode, on page 2210.

To return to user EXEC mode, use the following command:

Command	Purpose
`Device# disable`	Exits from privileged EXEC mode to user EXEC mode.

The following example shows the process of accessing privileged EXEC mode:

```
Device> enable
Password:<letmein>
Device#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes. To list the commands available in privileged EXEC mode, issue the **?** command at the prompt. From privileged EXEC mode you can access global configuration mode, which is described in the following section.

Note Because the privileged EXEC command set contains all of the commands available in user EXEC mode, some commands can be entered in either mode. In Cisco IOS XE documentation, commands that can be entered in either user EXEC mode or privileged EXEC mode are referred to as EXEC mode commands. If user or privileged is not specified in the documentation, assume that you can enter the referenced commands in either mode.

Global Configuration Mode

The term “global” is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal**privileged EXEC command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
`Device# configure terminal`	From privileged EXEC mode, enters global configuration mode.

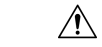
The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the host-name of the device followed by (config) and the pound sign (#). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example above, the system dialogue prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and “z” keys simultaneously; when you press these keys, **^Z** is printed to the screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.



Caution If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the **end** command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this only works in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose

`Device(config)# end`
or
`Device(config)# ^Z`
	Ends the current configuration session and returns to privileged EXEC mode.

`Device(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth or clock rate, refer to the Release 12.2 *Cisco IOS Interface Configuration Guide* . For protocol-specific commands, refer to the appropriate Cisco IOS XE software command reference.

To access and list the interface configuration commands, use the following command:

Command	Purpose
`Device(config)# interface type number`	Specifies the interface to be configured, and enters interface configuration mode.

In the following example, the user enters interface configuration mode for serial interface 0. The new prompt, *hostname* (config-if)#, indicates interface configuration mode.

```
Device(config)# interface serial 0
Device(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes entered from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements within the configuration mode. One example of a configuration submode is subinterface configuration mode, described in the following section.

Subinterface Configuration Mode

From interface configuration mode, you can enter subinterface configuration mode. Subinterface configuration mode is a submode of interface configuration mode. In subinterface configuration mode you can configure

multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols.

For detailed information on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS XE software documentation set.

To access subinterface configuration mode, use the following command in interface configuration mode:

Command	Purpose
`Device(config-if)# interface type number`	Specifies the virtual interface to be configured and enters subinterface configuration mode.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is identified as "2.1" to indicate that it is subinterface 1 of serial interface 2. The new prompt *hostname* (config-subif)# indicates subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Device(config)# interface serial 2
Device(config-if)# encapsulation frame-relay
Device(config-if)# interface serial 2.1
Device(config-subif)#
```

To exit subinterface configuration mode and return to interface configuration mode, use the **exit** command. To end your configuration session and return to privileged EXEC mode, press Ctrl-Z or enter the **end** command.

Cisco IOS XE CLI Sessions

Local CLI Sessions

Local CLI sessions require direct access to the the console port of the networking device. Local CLI sessions start in user EXEC mode. All of the tasks required to configure and manage a networking device can be done using a local CLI session. The most common method for establishing a local CLI session is to connect the serial port on a PC to the console port of the networking device and then to launch a terminal emulation application on the PC. The type of cable and connectors required and the settings for the terminal emulation application on the PC are dependant on the type of networking device that you are configuring. See to the documentation for your networking device for more information on setting it up for a local CLI session.

Remote CLI Sessions

Remote CLI sessions are created between a host such as a PC and a networking device such as a router over a network using a remote terminal access application such as Telnet and Secure Shell (SSH). Local CLI sessions start in user EXEC mode. Most of the tasks required to configure and manage a networking device can be done using a remote CLI session. The exceptions are tasks that interact directly with the console port (such as recovering from a corrupted operating system (OS) by uploading a new OS image over the console port) and interacting with the networking device when it is in ROM Monitor Mode.

This document explains how to configure security for remote Telnet sessions. Telnet is the most common method for accessing a remote CLI session on a networking device.

Note SSH is a more secure alternative to Telnet. SSH provides encryption for the session traffic between your local management device such as a PC and the networking device that you are managing. Encrypting the session traffic with SSH prevents hackers that might intercept the traffic from being able to decode it. See Secure Shell Version 2 Support feature module for more information on using SSH.

Terminal Lines are Used for Local and Remote CLI Sessions

Cisco networking devices use the word lines to refer to the software components that manage local and remote CLI sessions. You use the **line console 0** global configuration command to enter line configuration mode to configure options, such as a password, for the console port.

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# password password-string
```

Remote CLI sessions use lines that are referred to virtual teletypewriter (VTY) lines. You use the **line vty** *line-number* [*ending-line-number*] global configuration command to enter line configuration mode to configure options, such as a password, for remote CLI sessions.

```
Device# configure terminal
Device(config)# line vty 0 4
Device(config-line)# password password-string
```

Protect Access to Cisco IOS XE EXEC Modes

Cisco IOS XE provides the ability to configure passwords that protect access to the following:

Protecting Access to User EXEC Mode

The first step in creating a secure environment for your networking device is protecting access to user EXEC mode by configuring passwords for local and remote CLI sessions.

You protect access to user EXEC mode for local CLI sessions by configuring a password on the console port. See the Configuring and Verifying a Password for Local CLI Sessions, on page 2208.

You protect access to user EXEC mode for remote CLI sessions by configuring a password on the virtual terminal lines (VTYs). See the Configuring and Verifying a Password for Remote CLI Sessions, on page 2206 for instructions on how to configure passwords for remote CLI sessions.

Protecting Access to Privileged EXEC mode

The second step in creating a secure environment for your networking device is protecting access to privileged EXEC mode with a password. The method for protecting access to privileged EXEC mode is the same for local and remote CLI sessions.

You protect access to privileged EXEC mode by configuring a password for it. This is sometimes referred to as the enable password because the command to enter privileged EXEC mode is **enable**.

Command	Purpose

enable
 	
`Device> enable`
 	
`Password`
 	
`Device#`
	Enables privileged EXEC mode.
	• Enter your password if prompted. The password will not be shown in the terminal window.
	• The ">" at the end of the prompt string is changed to a "#" to indicate that you are in privileged EXEC mode.

Cisco IOS XE Password Encryption Levels

Some of the passwords that you configure on your networking device are saved in the configuration in plain text. This means that if you store a copy of the configuration file on a disk, anybody with access to the disk can discover the passwords by reading the configuration file. The following password types are stored as plain text in the configuration by default:

- Console passwords for local CLI sessions
- Virtual terminal line passwords for remote CLI sessions
- Username passwords using the default method for configuring the password
- Privileged EXEC mode password when it is configured with the **enable password** *password* command
- Authentication key chain passwords used by RIPv2 and EIGRP
- BGP passwords for authenticating BGP neighbors
- OSPF authentication keys for authenticating OSPF neighbors
- ISIS passwords for authenticating ISIS neighbors

This excerpt from a router configuration file shows examples of passwords and authentication keys that are stored as clear text.

```
!
enable password O9Jb6D
!
username username1 password 0 kV9sIj3
!
key chain trees
 key 1
  key-string willow
!
interface Ethernet1/0.1
 ip address 172.16.6.1 255.255.255.0
 ip router isis
 ip rip authentication key-chain trees
 ip authentication key-chain eigrp 1 trees
 ip ospf authentication-key j7876
 no snmp trap link-status
 isis password u7865k
!
line vty 0 4
 password V9jA5M
!
```

You can encrypt these clear text passwords in the configuration file by using the **service password-encryption** command. This should be considered only a minimal level of security because the encryption algorithm used by the **service password-encryption** command to encrypt passwords creates text strings that be decrypted using tools that are publicly available. You should still protect access to any electronic or paper copies of your configuration files after you use the **service password-encryption** command.

The **service password-encryption** command does not encrypt the passwords when they are sent to the remote device. Anybody with a network traffic analyzer who has access to you network can capture these passwords from the packets as they are transmitted between the devices. See the Configuring Password Encryption for Clear Text Passwords, on page 2212for more information on encrypting clear text passwords in configuration files.

Many of the Cisco IOS XE features that use clear text passwords can also be configured to use the more secure MD5 algorithm. The MD5 algorithm creates a text string in the configuration file that is much more difficult to decrypt. The MD5 algorithm does not send the password to the remote device. This prevents people using a traffic analyzer to capture traffic on your network from being able to discover your passwords.

You can determine the type of password encryption that has been used by the number that is stored with the password string in the configuration file of the networking device. The number 5 in the configuration excerpt below indicates that the enable secret password has been encrypted using the MD5 algorithm.

enable secret 5 1fGCS$rkYbR6.Z8xo4qCl3vghWQ0

The number 7 in the excerpt below indicates that the enable password has been encrypted using the less secure algorithm used by the **service password-encryption** command.

!

enable password 7 00081204

Cisco IOS XE CLI Session Usernames

After you have protected access to user EXEC mode and privileged EXEC mode by configuring passwords for them you can further increase the level of security on your networking device by configuring usernames to limit access to CLI sessions to your networking device to specific users.

Usernames that are intended to be used for managing a networking device can be modified with additional options such as:

See the *Cisco IOS Security Command Reference* . (http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html) for more information on how to configure the **username** command.

Cisco IOS XE Privilege Levels

The default configuration for Cisco IOS XE based networking devices uses privilege level 1 for user EXEC mode and privilege level 15 for privileged EXEC. The commands that can be run in user EXEC mode at privilege level 1 are a subset of the commands that can be run in privileged EXEC mode at privilege 15.

The **privilege** command is used to move commands from one privilege level to another. For example, some ISPs allow their first level technical support staff to enable and disable interfaces to activate new customer connections or to restart a connection that has stopped transmitting traffic. See the Example: Configuring a Device to Allow Users to Shutdown and Enable Interfaces, on page 2229 for an example of how to configure this option.

The **privilege** command can also be used to assign a privilege level to a username so that when a user logs in with the username, the session will run at the privilege level specified by the **privilege** command. For example if you want your technical support staff to view the configuration on a networking device to help them troubleshoot network problems without being able to modify the configuration, you can create a username, configure it with privilege level 15, and configure it to run the **show running-config** command automatically. When a user logs in with the username the running configuration will be displayed automatically. The user's session will be logged out automatically after the user has viewed the last line of the configuration. See the Example: Configuring a Device to Allow Users to View the Running Configuration, on page 2227 for an example of how to configure this option.

These command privileges can also be implemented when using AAA with TACACS+ and RADIUS. For example, TACACS+ provides two ways to control the authorization of router commands on a per-user or per-group basis. The first way is to assign privilege levels to commands and have the router verify with the TACACS+ server whether or not the user is authorized at the specified privilege level. The second way is to explicitly specify in the TACACS+ server, on a per-user or per-group basis, the commands that are allowed. For more information about implementing AAA with TACACS+ and RADIUS, see the technical note How to Assign Privilege Levels with TACACS+ and RADIUS .

Cisco IOS XE Password Configuration

Cisco IOS XE software does not prompt you to repeat any passwords that you configure to verify that you have entered the passwords exactly as you intended. New passwords, and changes to existing passwords, go into effect immediately after you press the Enter key at the end of a password configuration command string. If you make a mistake when you enter a new password and have saved the configuration on the networking device to its startup configuration file and exited privileged EXEC mode before you realize that you made a mistake, you may find that you are no longer able to manage the device.

The following are common situations that can happen:

- You make a mistake configuring a password for local CLI sessions on the console port.
  - If you have properly configured access to your networking device for remote CLI sessions, you can Telnet to it and reconfigure the password on the console port.
- You make a mistake configuring a password for remote Telnet or SSH sessions.
  - If you have properly configured access to your networking device for local CLI sessions, you can connect a terminal to it and reconfigure the password for the remote CLI sessions.
- You make a mistake configuring a password for privileged EXEC mode (enable password or enable secret password).
  - You will have to perform a lost password recovery procedure.
- You make a mistake configuring your username password, and the networking device requires that you log into it with your username.
  - If you do not have access to another account name, you will have to perform a lost password recovery procedure.

To protect yourself from having to perform a lost password recovery procedure open two CLI sessions to the networking device and keep one of them in privilege EXEC mode while you reset the passwords using the other session. You can use the same device (PC or terminal) to run the two CLI sessions or two different devices. You can use a local CLI session and a remote CLI session or two remote CLI sessions for this procedure. The CLI session that you use to configure the password can also be used to verify that the password

was changed properly. The other CLI session that you keep in privileged EXEC mode can be used to change the password again if you made a mistake the first time you configured it.

You should not save password changes that you have made in the running configuration to the startup configuration until you have verified that your password was changed successfully. If you discover that you made a mistake configuring a password, and you were not able to correct the problem using the second CLI session technique described above, you can power cycle the networking device so that it returns to the previous passwords that are stored in the startup configuration.

AES Password Encryption and Master Encryption Keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. To start using type-6 encryption, you must enable the AES password encryption feature and configure a master encryption key, which is used to encrypt and decrypt passwords. After you enable AES password encryption and configure a master key, all existing and newly created clear-text passwords for supported applications are stored in type-6 encrypted format, unless you disable type-6 password encryption. You can also configure the device to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Type 0 and type 7 passwords can be autoconverted to type 6 if the AES password encryption feature and master encryption key are configured.

Note Type 6 username and password are backward compatible to the Cisco IOS release 16.10.1 only. If you downgrade to any release version lower than Cisco IOS release 16.10.1, type 6 username and password will be rejected. After autoconversion, to avoid an administrator password getting rejected during a downgrade, migrate the passwords.

How To Configure Security with Passwords Privileges and Logins

Protecting Access to User Exec Mode

Configuring and Verifying a Password for Remote CLI Sessions

This task will assign a password for remote CLI sessions. After you have completed this task the networking device will prompt you for a password the next time that you start a remote CLI session with it.

Cisco IOS XE based networking devices require that you have a password configured for remote CLI sessions. If you attempt to start a remote CLI session with a device that doesn’t have a password configured for remote CLI sessions you will see a message that a password is required and has not been set. The remote CLI session will be terminated by the remote host.

Before you begin

If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal or a PC running a terminal emulation application, attached to the console port.

Your terminal, or terminal emulation application, must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

To perform the verification step (Step 6) for this task, your networking device must have an interface that is in an operational state. The interface must have a valid IP address.

Note If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal attached to the console port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *line-number* [*ending-line-number*]
4. **password** *password*
5. **end**
6. **telnet** *ip-address*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line vty** *line-number* [*ending-line-number*] **Example:** `Device(config)# line vty 0 4`	Enters line configuration mode.

	Command or Action	Purpose
Step 4	**password** *password* **Example:** `Device(config-line)# password H7x3U8`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Device(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**telnet** *ip-address* **Example:** `Device# telnet 172.16.1.1`	Start a remote CLI session with the networking device from your current CLI session using the IP address of an interface in the networking device that is in an operational state (interface up, line protocol up). • Enter the password that you configured is step 4 when prompted. **Note** This procedure is often referred to as a starting a recursive Telnet session because you are initiating a remote Telnet session with the networking device from the networking device itself.
Step 7	**exit** **Example:** `Device# exit`	Terminates the remote CLI session (recursive Telnet session) with the networking device.

Troubleshooting Tips

To display information for all users who have access to a lawful intercept view, issue the **show users lawful-intercept** command. (This command is available only to authorized lawful intercept view users.)

What to Do Next

Proceed to the Configuring and Verifying a Password for Local CLI Sessions, on page 2208.

Configuring and Verifying a Password for Local CLI Sessions

This task will assign a password for local CLI sessions over the console port. After you have completed this task, the networking device will prompt you for a password the next time that you start a local CLI session on the console port.

This task can be performed over a local CLI session using the console port or a remote CLI session. If you want to perform the optional step of verifying that you configured the password correctly you should perform this task using a local CLI session using the console port.

Before you begin

If you want to perform the optional step of verifying the local CLI session password, you must perform this task using a local CLI session. You must have a terminal or a PC running a terminal emulation program, connected to the console port of the networking device. Your terminal must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line console 0**
4. **password** *password*
5. **end**
6. **exit**
7. Press the Enter key.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line console 0** **Example:** `Device(config)# line console 0`	Enters line configuration mode and selects the console port as the line that you are configuring.
Step 4	**password** *password* **Example:** `Device(config-line)# password Ji8F5Z`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot

	Command or Action	Purpose
		specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Device(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.
Step 7	Press the Enter key.	(Optional) Initiates the local CLI session on the console port. • Enter the password that you configured is step 4 when prompted to verify that it was configured correctly. **Note** This step can be performed only if you are using a local CLI session to perform this task.

Troubleshooting Tips

If your new password is not accepted proceed to the Configuration Examples for Configuring Security with Passwords Privileges and Logins for instructions on what to do next.

What to Do Next

Proceed to the Protecting Access to Privileged EXEC Mode, on page 2210.

Protecting Access to Privileged EXEC Mode

Configuring and Verifying the Enable Password

Cisco no longer recommends that you use the **enable password** command to configure a password for privileged EXEC mode. The password that you enter with the **enable password** command is stored as plain text in the configuration file of the networking device. You can encrypt the password for the **enable password** command in the configuration file of the networking device using the **service password-encryption** command. However the encryption level used by the **service password-encryption** command can be decrypted using tools available on the Internet.

Instead of using the **enable password**command, Cisco recommends using the **enable secret** commandbecause it encrypts the password that you configure with it with strong encryption . For more information on password encryption issues see the Cisco IOS XE Password Encryption Levels, on page 2203. For information on configuring the **enable secret**command see the Configuring and Verifying the Enable Secret Password, on page 2213.

Note The networking device must not have a password configured by the **enable secret** command in order to perform this task successfully. If you have already configured a password for privileged EXEC mode using the **enable secret** command, the password configured takes precedences over the password that you configure in this task using the **enable password** command.

You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable password** *password*
4. **end**
5. **exit**
6. **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**enable password** *password*
	Example:
	`Device(config)# enable password t6D77CdKq`
		The argument *password* is a character string that specifies the enable password. The following rules apply to the *password* argument:
		• Must contain from 1 to 25 uppercase and lowercase alphanumeric characters.
		• Must not have a number as the first character.
		• Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized.
		• Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following:
		  • Enter abc
		  • Type Crtl-v

	Command or Action	Purpose
		• Enter ?123
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 5	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.
Step 6	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter the password you configured in step 3.

Troubleshooting Tips

If your new password is not accepted, proceed to the Recovering from Lost or Misconfigured Passwords for Privileged EXEC Mode section for instructions on what to do next.

What to Do Next

Encrypt the clear text enable password in the configuration file of the networking device using the procedure described in Configuring Password Encryption for Clear Text Passwords, on page 2212.

Configuring Password Encryption for Clear Text Passwords

Cisco IOS XE stores passwords in clear text in network device configuration files for several features such as passwords for local and remote CLI sessions, and passwords for neighbor authentication for routing protocols. Clear text passwords are a security risk because anybody with access to archived copies of the configuration files can discover the passwords that are stored as clear text. The **service password-encryption** command can be used to encrypt clear text commands in the configuration files of networking devices. See the Cisco IOS XE Password Encryption Levels, on page 2203 for more information.

Perform the following steps to configure password encryption for passwords that are stored as clear text in the configuration files of your networking device.

Before you begin

You must have at least one feature that uses clear text passwords configured on your networking device for this command to have any immediate effect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service password-encryption**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service password-encryption** **Example:** `Device(config)# service password-encryption`	Enables Password encryption for all passwords clear text passwords, including username passwords, authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol neighbor passwords.
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuring and Verifying the Enable Secret Password

Cisco recommends that you use the **enable secret** command, instead of the **enable** *password* command to configure a password for privileged EXEC mode. The password created by the **enable secret** command is encrypted with the more secure MD5 algorithm.

Note You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Perform one of the following steps:
   - **enable secret** *password*
   - **enable secret 5** *previously-encrypted-password*
4. **end**
5. **exit**
6. **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Perform one of the following steps:
	• **enable secret** *password*
	• **enable secret 5** *previously-encrypted-password*
	Example:
	`Device(config)# enable secret t6D77CdKq`
	Example:
	`Device(config)# enable secret 5 $1$/x6H$RhnDI3yLC4GA01aJnHLQ4/`
		The argument *password* is a character string that specifies the **enable secret** password. The following rules apply to the *password* argument:
		• Must contain from 1 to 25 uppercase and lowercase alphanumeric characters.
		• Must not have a number as the first character.
		• Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized.
		• Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following:
		• Enter abc
		• Type Crtl-v
		• Enter ?123
		or
		Sets a previously encrypted password for privileged EXEC mode by entering the number 5 before the previously encrypted string. You must enter an exact copy of a password from a configuration file that was previously encrypted by the **enable secret** command to use this method.

Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 5	**exit** **Example:**	Exits privileged EXEC mode.

	Command or Action	Purpose
	`Device# exit`	
Step 6	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter the password that you configured in Step 3.

Troubleshooting Tips

If your new password is not accepted proceed to the Configuration Examples for Configuring Security with Passwords Privileges and Logins for instructions on what to do next.

What to Do Next

If you have finished configuring passwords for local and remote CLI sessions and you want to configure additional security features, such as usernames, and privilege levels proceed to the Configuring Security Options to Manage Access to CLI Sessions and Commands, on page 2217.

Configuring a Device to Allow Users to View the Running Configuration

To access the running configuration of a device using the **show running-config** command at a privilege level lower than level 15, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **privilege exec all level** *level command-string*
4. **file privilege** *level*
5. **privilege configure all level** *level command-string*
6. **end**
7. **show privilege**
8. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**privilege exec all level** *level command-string* **Example:** `Device(config)# privilege exec all level 5 show running-config`	Changes the privilege level of the specified command from one privilege level to another.
Step 4	**file privilege** *level* **Example:** `Device(config)# file privilege 5`	Allows a user of the privilege level to execute commands that involve the file system on a device.
Step 5	**privilege configure all level** *level command-string* **Example:** `Device(config)# privilege configure all level 5 logging`	Allows a user of a privilege level to see specific configuration commands. For example, allows the user of privilege level 5 to see the logging configuration commands in the running configuration.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**show privilege** **Example:** `Device# show privilege`	Displays the current privilege level.
Step 8	**show running-config** **Example:** `Device# show running-config`	Displays the current running configuration for the specified privilege level.

Example

The following output for the **show running-config** command displays the logging configuration commands in the running configuration. Users with a privilege level below 15 can view the running configuration after configuring the **privilege configure all level** *level command-string* command.

```
Device# show running-config

Building configuration...

Current configuration : 128 bytes
!
boot-start-marker
boot-end-marker
!
no logging queue-limit
logging buffered 10000000
```

```
no logging rate-limit
!
!
!
end
```

Configuring Security Options to Manage Access to CLI Sessions and Commands

The tasks in this section describe how to configure your networking device to permit the use of a subset of privileged EXEC mode commands by users who should not have access to all of of the commands available in privileged EXEC mode.

These tasks are beneficial for companies that have multiple levels of network support staff and the company wants the staff at each level to have access to a different subset of the privileged EXEC mode commands.

In this task the users who should not have access to all of of the commands available in privileged EXEC mode are referred to as the first-line technical support staff.

This section contains the following procedures:

Configuring the Networking Device for the First-Line Technical Support Staff

This task describes how to configure the networking device for first-line technical support users. First-line technical support staff are usually not allowed to run all of the commands available in privileged EXEC mode (privilege level 15) on a networking device. They are prevented from running commands that they are not authorized for by not being granted access to the password assigned to privileged EXEC mode or to other roles that have been configured on the networking device.

The **privilege** command is used to move commands from one privilege level to another in order to create the additional levels of administration of a networking device that is required by companies that have different levels of network support staff with different skill levels.

The default configuration of a Cisco IOS XE device permits two types of users to access the CLI. The first type of user is a person who is only allowed to access user EXEC mode. The second type of user is a person who is allowed access to privileged EXEC mode. A user who is only allowed to access user EXEC mode is not allowed to view or change the configuration of the networking device, or to make any changes to the operational status of the networking device. On the other hand, a user who is allowed access to privileged EXEC mode can make any change to a networking device that is allowed by the CLI.

In this task the two commands that normally run at privilege level 15 are reset to privilege level 7 using the privilege command in order that first-line technical support users will be allowed to run the two commands. The two commands for which the privilege levels will be reset are the **clear counters** command and **reload** command.

- The **clear counters** command is used to reset the counter fields on interfaces for statistics such as packets received, packets transmitted, and errors. When a first-line technical support user is troubleshooting an interface related connectivity issue between networking devices, or with remote users connecting to the network, it is useful to reset the interface statistics to zero and them monitor the interfaces for a period of time to see if the values in the interface statistics counters change.
- The **reload** command is used initiate a reboot sequence for the networking device. One common use of the reload command by first-line technical support staff is to cause the networking device to reboot during a maintenance window so that it loads a new operating system that was previously copied onto the networking device's file system by a user with a higher level of authority.

Any user that is permitted to know the **enable secret** password that is assigned to the first-line technical support user role privilege level can access the networking device as a first-line technical support user. You can add an additional level of security by configuring a username on the networking device and requiring that the users know the username and the password. Configuring a username as an additional level of security is described in the . Configuring a Device to Require a Username for the First-Line Technical Support Staff, on page 2221

Note You must not have the **aaa new-model** command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown i steps in this task. See the Cisco IOS command reference book for your Cisco IOS release fo on the additional arguments and keywords that can be used with these commands.

Caution Do not use the no form of the **privilege** command to reset the privilege level of a comm because it might not return the configuration to the correct default state. Use the **reset** keyw command instead to return a command to its default privilege level. For example, to rem **exec level** *reload* command from the configuration and return the **reload** command to it 15, use the **privilege exec reset** *reload* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable secret level** *level password*
4. **privilege exec level** *level command-string*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. Enter the password when prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**enable secret level** *level password* **Example:**	Configures a new enable secret password for privilege level 7.

	Command or Action	Purpose
	`Device(config)# enable secret level 7 Zy72sKj`	
Step 4	**privilege exec level** *level command-string* **Example:** `Device(config)# privilege exec level 7 clear counters`	Changes the privilege level of the **clear counters** command from privilege level 15 to privilege level 7.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

Verifying the Configuration for the First-Line Technical Support Staff

This task describes how to verify that the network device is configured correctly for the first-line technical support staff.

Before you begin

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

SUMMARY STEPS

1. **enable** *level password*
2. **show privilege**
3. **clear counters**
4. **clear ip route** *
5. **reload in time**
6. **reload cancel**
7. **disable**
8. **show privilege**

DETAILED STEPS

Step 1 **enable** *level password*

Logs the user into the networking device at the privilege level specified for the level argument.

Example:

```
Device> enable 7 Zy72sKj
```

Step 2 **show privilege**

Displays the privilege level of the current CLI session

Example:

```
Device# show privilege

Current privilege level is 7
```

Step 3 **clear counters**

The clear counters command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

Example:

```
Device# clear counters

Clear "show interface" counters on all interfaces [confirm]
Device#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

Step 4 **clear ip route ***

The *ip route* argument string for the **clear** command should not be allowed because it was not changed from privilege level 15 to privilege level 7.

Example:

```
Device# clear ip route *
                                                                        ^
% Invalid input detected at '^' marker.
```

Step 5 **reload in time**

The reload command causes the networking device to reboot.

Example:

```
Device# reload in

10
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]

Device#

***
*** --- SHUTDOWN in 0:10:00 ---
***
02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
```

Step 6 **reload cancel**

The **reload cancel** terminates a reload that was previously setup with the the **reload in** *time* command.

Example:

```
Device# reload cancel

***
*** --- SHUTDOWN ABORTED ---
***
04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload cancelled at 15:38:46 PST Sun Mar 27
 2005
```

Step 7 **disable**

Exits the current privilege level and returns to privilege level 1.

Example:

```
Device# disable
```

Step 8 **show privilege**

Displays the privilege level of the current CLI session

Example:

```
Device> show privilege

Current privilege level is 1
```

Troubleshooting Tips

If your configuration does not work the way that you want it to and you want to remove the privilege commands from the configuration, use the **reset** keyword for the **privilege** command to return the commands to their default privilege level. For example, to remove the command **privilege exec level** *reload* command from the configuration and return the **reload** command to its default privilege of 15 use the **privilege exec reset** *reload*command.

What to Do Next

If you want to add an additional level of security by requiring that the first level technical staff use a login name, proceed to the Configuring a Device to Require a Username for the First-Line Technical Support Staff, on page 2221.

Configuring a Device to Require a Username for the First-Line Technical Support Staff

This task configures the networking device to require that the first-line technical support staff login to the networking device with a login name of admin. The admin username configured in this task is assigned the privilege level of 7 which will allow users who log in with this name to run the commands that were reassigned to privilege level 7 in the previous task. When a user successfully logs in with the admin username, the CLI session will automatically enter privilege level 7.

Before Cisco IOS XE Release 2.3, two types of passwords were associated with usernames: Type 0, which is a clear text password visible to any user who has access to privileged mode on the router, and type 7, which has a password encrypted by the **service password encryption** command.

In Cisco IOS XE Release 2.3 and later releases, the new **secret** keyword for the **username** command allows you to configure Message Digest 5 (MD5) encryption for username passwords.

Before you begin

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

See the Configuring the Networking Device for the First-Line Technical Support Staff, on page 2217 for instructions on how to change the privilege level for a command.

Note MD5 encryption for the **username** command is not supported in versions of Cisco IOS software prior to Cisco IOS XE Release 2.3.

You must not have the aaa-new model command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown i steps in this task. Refer to the Cisco IOS command reference book for your Cisco IOS X information on the additional arguments and keywords that can be used with these com

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **username** *username* **privilege** *level* **secret** *password*
4. **end**
5. **disable**
6. **login** *username*
7. **show privilege**
8. **clear counters**
9. **clear** *ip route* *
10. **reload in** *time*
11. **reload cancel**
12. **disable**
13. **show privilege**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enters privileged EXEC mode. Enter the password when prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**username** *username* **privilege** *level* **secret** *password* **Example:** `Device(config)# username admin privilege 7 secret Kd65xZa`	Creates a username and applies MD5 encryption to the *password* text string.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 5	**disable** **Example:** `Device# disable`	Exits the current privilege level and returns to user EXEC mode.
Step 6	**login** *username* **Example:** `Device> login admin`	Logs in the user. Enter the username and password you configured in step 3 when prompted.

Step 7	**show privilege**
	Example:
	`Device# show privilege`
	`Current privilege level is 7`
		The **show privilege** command displays the privilege level of the CLI session.

Step 8	**clear counters**
	Example:
	`Device# clear counters`
	`Clear "show interface" counters on all interfaces [confirm]`
	`Device#`
	`02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console`
		The **clear counters** command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

	Command or Action	Purpose

Step 9	**clear** *ip route* *
	 Example:
	 `Device# clear ip route *`
	 `                     ^`
	 `% Invalid input detected at '^' marker.`
		The *ip route* argument string for the **clear** command is not allowed because it was not changed from privilege level 15 to privilege level 7.

Step 10	**reload in** *time*
	 Example:
	 `Device# reload in`
	 `10`
	 `Reload scheduled in 10 minutes by console`
	 `Proceed with reload? [confirm]`
	 `Device#`
	 `***`
	 `*** --- SHUTDOWN in 0:10:00 ---`
	 `***`
	 `02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20`
		The reload command causes the networking device to reboot.

Step 11	**reload cancel**
	 Example:
	 `Device# reload cancel`
	 `***`
	 `*** --- SHUTDOWN ABORTED ---`
	 `***`
	 `04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED: Scheduled reload cancelled at 15:38:46 PST Sun Mar 27 2005`
		The **reload cancel** command terminates a reload that was previously setup with the the **reload in** *time* command.

Step 12	**disable** **Example:** `Device# disable`	Exits the current privilege level and returns to user EXEC mode.

Step 13	**show privilege**
	 Example:
	 `Device> show privilege`
	 `Current privilege level is 1`
		Displays the privilege level of the current CLI session

Recovering from a Lost or Misconfigured Password for Local Sessions

There are three methods that can be used to recover from a lost or misconfigured password for local CLI sessions over console port. The method that you will use depends on the current configuration of your networking device.

Networking Device Is Configured to Allow Remote CLI Sessions

The fastest method to recover from a lost, or misconfigured password for local CLI sessions is to establish a remote CLI session with the networking device and repeat the Configuring and Verifying a Password for Local CLI Sessions, on page 2208. Your networking device must be configured to allow remote CLI sessions and you must know the remote CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Remote CLI Sessions

- If you cannot establish a remote session to your networking device, and you have not saved the misconfigured local CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous local CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Recovering from a Lost or Misconfigured Password for Remote Sessions

There are three methods that can be used to recover from a lost, or misconfigured remote CLI session password. The method that you will use depends on the current configuration of your networking device.

Networking Device Is Configured to Allow Local CLI Sessions

The fastest method to recover from a lost, or misconfigured password for remote CLI sessions is to establish a local CLI session with the networking device and repeat the Configuring and Verifying a Password for Remote CLI Sessions, on page 2206. Your networking device must be configured to allow local CLI sessions and you must know the local CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Local CLI Sessions

- If you cannot establish a local CLI session to your networking device, and you have not saved the misconfigured remote CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous remote CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Recovering from Lost or Misconfigured Passwords for Privileged EXEC Mode

There are two methods that can be used to recover from a lost, or misconfigured Privileged EXEC Mode password. The method that you will use depends on the current configuration of your networking device.

A Misconfigured Privileged EXEC Mode Password Has Not Been Saved

- If you have not saved the misconfigured privileged EXEC mode password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous privileged EXEC mode password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Configuration Examples for Configuring Security with Passwords Privileges and Logins

Example: Configuring an Encrypted Preshared Key

The following is an example of a configuration for which a type 6 preshared key has been encrypted. It includes the prompts and messages that a user might see.

```
Device(config)# password encryption aes
New key:
Confirm key:
Device (config)#

01:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Device (config)# exit
```

Example: Configuring a Device to Allow Users to Clear Remote Sessions

The following example shows how to configure a networking device to allow a non administrative user to clear remote CLI session virtual terminal (VTY) lines.

The first section is an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
privilege exec level 7 clear line
!
no aaa new-model
!
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
```

```
!
privilege exec level 7 clear line
!
! the privilege exec level 7 clear command below is entered automatically
! when you enter the privilege exec level 7 clear line command above, do
! not enter it again
!
privilege exec level 7 clear
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege

Current privilege level is 7
R1#
```

The following section using the **show user** command shows that two users (admin and root) are currently logged in to the networking device:

```
R1# show user

    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00
   2 vty 0     root       idle                 00:00:17 172.16.6.2
  Interface    User               Mode         Idle     Peer Address
```

The following section using the **clear line** *2* command terminates the remote CLI session in use by the username root:

```
R1# clear line  2

[confirm]
 [OK]
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00
  Interface    User               Mode         Idle     Peer Address
```

Example: Configuring a Device to Allow Users to View the Running Configuration

For Users With Privilege Level 15

The following example shows how to configure the networking device to allow a non administrative users (no access to privileged EXEC mode) to view the running configuration automatically. This example requires

that the username is configured for privilege level 15 because many of the commands in the configuration file can be viewed only by users who have access to privilege level 15.

The solution is to temporarily allow the user access to privilege level 15 while running the **show running-config** command and then terminating the CLI session when the end of the configuration file has been viewed. In this example the networking device will automatically terminate the CLI session when the end of the configuration file has been viewed. No further configuration steps are required.

Caution You must include the **noescape** keyword for the **username** command to prevent the user from entering an escape character that will terminate viewing the configuration file and leave the session running at privilege level 15.

```
!
!
username viewconf privilege 15 noescape secret 5 $1$zA9C$TDWD/Q0zwp/5xRwRqdgc/.
username viewconf autocommand show running-config
!
```

For Users With Privilege Level Lower Than Level 15

The following example shows how to configure a networking device to allow a user with privilege level lower than level 15 to view the running configuration.

```
Device> enable
Device# configure terminal
Device(config)# privilege exec all level 5 show running-config
Device(config)# file privilege 5
Device(config)# privilege configure all level 5 logging
Device(config)# end
Device# show privilege

Current privilege level is 5

Device# show running-config

Building configuration...

Current configuration : 128 bytes
!
boot-start-marker
boot-end-marker
!
no logging queue-limit
logging buffered 10000000
no logging rate-limit
!
!
!
end
```

Example: Configuring a Device to Allow Users to Shutdown and Enable Interfaces

The following example shows how to configure a networking device to allow non administrative users to shutdown and enable interfaces.

The first section is an an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
no aaa new-model
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
!
privilege interface all level 7 shutdown
privilege interface all level 7 no shutdown
privilege configure level 7 interface
privilege exec level 7 configure terminal
!
! the privilege exec level 7 configure command below is entered automatically
! when you enter the privilege exec level 7 configure terminal command above, do
! not enter it again
!
privilege exec level 7 configure
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege
Current privilege level is 7
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00
  Interface    User               Mode         Idle     Peer Address
```

The following section shows that the admin user is permitted to shutdown and enable an interface:

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface ethernet 1/0
R1(config-if)# shutdown
R1(config-if)# no shutdown
R1(config-if)# exit
R1#
```

Where to Go Next

Once you have established a baseline of security for your networking devices you can consider more advanced options such as:

- Role-Based CLI Access--The role-based CLI access feature offers a more comprehensive set of options than the **privilege** command (described in this document) for network managers who want to allow different levels of technical support staff to have different levels of access to CLI commands.
- AAA Security--Many Cisco networking devices offer an advanced level of security using authentication, authorization and accounting (AAA) features. All of the tasks described in this document, and other - more advanced security features - can be implemented using AAA on the networking device in conjunction with a remote TACACS+ or RADIUS server. For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Cisco IOS XE Security Configuration Guide:Securing User Services* , Release 2.

Additional References

The following sections provide references related to Configuring Security with Passwords and, Login Usernames for CLI Sessions on Networking Devices.

Related Documents

Related Topic	Document Title
Managing user access to CLI commands and configuration information	"Role-Based CLI Access" in the *Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
AAA Security Features	*Cisco IOS XE Security Configuration Guide: Securing User Services* , Release 2
Assigning privilege levels with TACACS+ and RADIUS	How to Assign Privilege Levels with TACACS+ and RADIUS

Standards

Standard	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Security with Passwords Privileges and Logins

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 224: Feature Information for Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices

Feature Name	Releases	Feature Configuration Information
Enhanced Password Security		Using the Enhanced Password Security feature, you can configure MD5 encryption for username passwords. MD5 encryption is a one-way hash function that makes reversal of an encrypted password impossible, providing strong encryption protection. Using MD5 encryption, you cannot retrieve clear text passwords. MD5 encrypted passwords cannot be used with protocols that require that the clear text password be retrievable, such as Challenge Handshake Authentication Protocol (CHAP).

CHAPTER 170

Role-Based CLI Access

The Role-Based CLI Access feature allows the network administrator to define views, which are a set of operational commands and configuration capabilities that provide selective or partial access to Cisco IOS EXEC and configuration (config) mode commands. Views restrict user access to Cisco IOS command-line interface (CLI) and configuration information; that is, a view can define what commands are accepted and what configuration information is visible. Thus, network administrators can exercise better control over access to Cisco networking devices.

Prerequisites for Role-Based CLI Access

Your image must support CLI views.

Restrictions for Role-Based CLI Access

Lawful Intercept Images Limitation

CLI views are a part of all platforms and Cisco IOS images because they are a part of the Cisco IOS parser. However, the lawful intercept view is available only in images that contain the lawful intercept subsystem.

Maximum Number of Allowed Views

The maximum number of CLI views and superviews, including one lawful intercept view, that can be configured is 15. (This does not include the root view.)

Parse View Profiles

When you configure Parse View profiles, the 'no' or 'default' commands in combination with any configuration commands are not saved to the startup-configuration file. The configuration is accepted and is persistent until the device is reloaded. Examples of commands which are not saved to the startup-configuration:

- **command configure include all no**
- **command interface include all no**
- **command configure include all default**

Information About Role-Based CLI Access

Benefits of Using CLI Views

Although users can control CLI access via both privilege levels and enable mode passwords, these functions do not provide network administrators with the necessary level of detail needed when working with Cisco IOS devices. CLI views provide a more detailed access control capability for network administrators, thereby, improving the overall security and accountability of Cisco IOS software.

As of Cisco IOS Release 12.3(11)T, network administrators can also specify an interface or a group of interfaces to a view; thereby, allowing access on the basis of specified interfaces.

Root View

When a system is in root view, it has all of the access privileges as a user who has level 15 privileges. If the administrator wishes to configure any view to the system (such as a CLI view, a superview, or a lawful intercept view), the system must be in root view.

The difference between a user who has level 15 privileges and a root view user is that a root view user can configure a new view and add or remove commands from the view. Also, when you are in a CLI view, you have access only to the commands that have been added to that view by the root view user.

Lawful Intercept View

Like a CLI view, a lawful intercept view restricts access to specified commands and configuration information. Specifically, a lawful intercept view allows a user to secure access to lawful intercept commands that are held within the TAP-MIB, which is a special set of simple network management protocol (SNMP) commands that store information about calls and users.

Commands available in lawful intercept view belong to one of the these categories:

- Lawful intercept commands that should not be made available to any other view or privilege level
- CLI views that are useful for lawful intercept users but do not have to be excluded from other views or privilege levels

Superview

A superview consists of one or more CLI views, which allow users to define what commands are accepted and what configuration information is visible. Superviews allow a network administrator to easily assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

Superviews contain these characteristics:

- A CLI view can be shared among multiple superviews.
- Commands cannot be configured for a superview; that is, you must add commands to the CLI view and add that CLI view to the superview.
- Users who are logged into a superview can access all of the commands that are configured for any of the CLI views that are part of the superview.
- Each superview has a password that is used to switch between superviews or from a CLI view to a superview.
- If a superview is deleted, its associated CLI views are not deleted.

View Authentication via a New AAA Attribute

View authentication is performed by an external authentication, authorization, and accounting (AAA) server via the new attribute **cli-view-name**.

AAA authentication associates only one view name to a particular user; that is, only one view name can be configured for a user in an authentication server.

How to Use Role-Based CLI Access

Configuring a CLI View

Perform this task to create a CLI view and add commands or interfaces to the view, as appropriate.

Before you begin

Before you create a view, you must perform the following tasks:

- Enable AAA using the **aaa new-model** command.
- Ensure that your system is in root view-not privilege level 15.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *view-name* [**inclusive**]
4. **secret** [**0** | **5**] *encrypted-password*

5. **commands** *parser-mode* {**exclude** | **include-exclusive** | **include**} [**all**] [**interface** *interface-name* | *command*]
6. **end**
7. **enable** [*privilege-level* | **view** *view-name*]
8. **show parser view all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Device> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**parser view** *view-name* [**inclusive**]
	Example:
	`Device(config)# parser view first inclusive`
	`Device(config-view)#`
		Creates a view including all commands by default. If the **inclusive** keyword option is not selected, it creates a view excluding all commands by default. You are in the view configuration mode.

Step 4	**secret** [**0** | **5**] *encrypted-password* **Example:** `Device(config-view)# secret 5 secret`	Associates a CLI view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view. **Note** With CSCts50236, the password can be removed or overwritten. Use the **no secret** command to remove the configured password.
Step 5	**commands** *parser-mode* {**exclude** | **include-exclusive** | **include**} [**all**] [**interface** *interface-name* | *command*] **Example:** `Device(config-view)# commands exec include show version`	Adds commands or interfaces to a view and specifies the mode in which the specified command exists. **Note** While configuring parser view profiles, the following **no** or **default** commands are not saved to the startup configuration. These commands are in use until the device is reloaded. Once the device is reloaded, reapply these commands to get the required results. • **commands configure include all no** • **commands interface include all no** • **commands configure include all default**

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-view)# end`	Exits view configuration mode and returns to privileged EXEC mode.
Step 7	**enable** [*privilege-level* | **view** *view-name*] **Example:** `Device# enable view first`	Prompts you for a password to access a configured CLI view, and you can switch from one view to another view. Enter the password to access the CLI view.
Step 8	**show parser view all** **Example:** `Device# show parser view all`	(Optional) Displays information for all views that are configured on the device. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Troubleshooting Tips

You must associate a password with a view. If you do not associate a password, and you attempt to add commands to the view using the **commands** command, a system message such as the following is displayed:

```
%Password not set for view <viewname>.
```

Configuring a Lawful Intercept View

Perform this task to initialize and configure a view for lawful-intercept-specific commands and configuration information.

Before you begin

Before you initialize a lawful intercept view, ensure that the privilege level is set to 15 using the **privilege** command.

Note Only an administrator or a user who has level 15 privileges can initialize a lawful intercept view.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **li-view** *li-password* **user** *username* **password** *password*
4. **username lawful-intercept** [*name*] [**privilege** *privilege-level* | **view** *view-name*] **password** *password*

5. **parser view** *view-name*
6. **secret 5** *encrypted-password*
7. **name** *new-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Device> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**li-view** *li-password* **user** *username* **password** *password* **Example:** `Device(config)# li-view lipass user li_admin password li_adminpass`	Initializes a lawful intercept view. After the li-view is initialized, you must specify at least one user via **user** *username* **password** *password* options.
Step 4	**username lawful-intercept** [*name*] [**privilege** *privilege-level* | **view** *view-name*] **password** *password* **Example:** `Device(config)# username lawful-intercept li-user1 password li-user1pass`	Configures lawful intercept users on a Cisco device.
Step 5	**parser view** *view-name* **Example:** `Device(config)# parser view li view name`	(Optional) Enters view configuration mode, which allows you to change the lawful intercept view password or the lawful intercept view name.
Step 6	**secret 5** *encrypted-password* **Example:** `Device(config-view)# secret 5 secret`	(Optional) Changes an existing password for a lawful intercept view.
Step 7	**name** *new-name* **Example:** `Device(config-view)# name second`	(Optional) Changes the name of a lawful intercept view. If this command is not issued, the default name of the lawful intercept view is “li-view.”

Troubleshooting Tips

To display information for all users who have access to a lawful intercept view, issue the **show users lawful-intercept** command. (This command is available only to authorized lawful intercept view users.)

Configuring a Superview

Perform this task to create a superview and add at least one CLI view to the superview.

Before you begin

Before adding a CLI view to a superview, ensure that the CLI views that are added to the superview are valid views in the system; that is, the views have been successfully created using the **parser view** command.

Note You can add a view to a superview only after you configure a password for the superview (using the **secret 5** command). Thereafter, issue the **view** command in view configuration mode to add at least one CLI view to the superview.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *superview-name* **superview**
4. **secret 5** *encrypted-password*
5. **view** *view-name*
6. **end**
7. **show parser view all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Device> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parser view** *superview-name* **superview** **Example:** `Device(config)# parser view su_view1 superview`	Creates a superview and enters view configuration mode.
Step 4	**secret 5** *encrypted-password* **Example:** `Device(config-view)# secret 5 secret`	Associates a CLI view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view.
Step 5	**view** *view-name*	Adds a normal CLI view to a superview.

	Command or Action	Purpose
	Example: `Device(config-view)# view view_three`	Issue this command for each CLI view that is to be added to a given superview.

Step 6	**end**
	Example:
	
	`Device(config-view)# end`
	`Device#`
		Exits view configuration mode and returns to privileged EXEC mode.

Step 7	**show parser view all** **Example:** `Device# show parser view`	(Optional) Displays information for all views that are configured on the device. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Monitoring Views and View Users

To display debug messages for all views-root, CLI, lawful intercept, and superview-use the **debug parser view** command in privileged EXEC mode.

Configuration Examples for Role-Based CLI Access

Example: Configuring a CLI View

The following example shows how to configure two CLI views, “first” and “second". Thereafter, you can verify the CLI view in the running configuration.

```
Device(config)# parser view first inclusive
Device(config-view)# secret 5 firstpass
Device(config-view)# command exec exclude show version
Device(config-view)# command exec exclude configure terminal
Device(config-view)# command exec exclude all show ip
Device(config-view)# exit
Device(config)# parser view second
Device(config-view)# secret 5 secondpass
Device(config-view)# command exec include-exclusive show ip interface
Device(config-view)# command exec include logout
Device(config-view)# exit
!
!
Device(config-view)# do show running-config | beg view

parser view first inclusive
 secret 5 $1$MCmh$QuZaU8PIMPlff9sFCZvgW/
```

```
 commands exec exclude configure terminal
 commands exec exclude configure
 commands exec exclude all show ip
 commands exec exclude show version
 commands exec exclude show
!
parser view second
 secret 5 $1$iP2M$R16BXKecMEiQesxLyqygW.
 commands exec include-exclusive show ip interface
 commands exec include show ip
 commands exec include show
 commands exec include logout
!
```

Example: Verifying a CLI View

After you have configured the CLI views “first” and “second", you can issue the **enable view**command to verify which commands are available in each view. The following example shows which commands are available inside the CLI view “first” after the user has logged into this view. (Because the **show ip** command is configured with the all option, a complete set of suboptions is shown, except the **show ip interface** command, which is using the **include-exclusive** keyword in the second view.)

```
Device# enable view first
Password:
Device# ?
Exec commands:
  configure  Enter configuration mode
  enable     Turn on privileged commands
  exit       Exit from the EXEC
  show       Show running system information
Device# show ?
  ip       IP information
  parser   Display parser information
  version  System hardware and software status
Device# show ip ?

  access-lists            List IP access lists
  accounting              The active IP accounting database
  aliases                 IP alias table
  arp                     IP ARP table
  as-path-access-list     List AS path access lists
  bgp                     BGP information
  cache                   IP fast-switching route cache
  casa                    display casa information
  cef                     Cisco Express Forwarding
  community-list          List community-list
  dfp                     DFP information
  dhcp                    Show items in the DHCP database
  drp                     Director response protocol
  dvmrp                   DVMRP information
  eigrp                   IP-EIGRP show commands
  extcommunity-list       List extended-community list
  flow                    NetFlow switching
  helper-address          helper-address table
  http                    HTTP information
  igmp                    IGMP information
  irdp                    ICMP Device Discovery Protocol
.
.
.
```

Example: Configuring a Lawful Intercept View

The following example shows how to configure a lawful intercept view, add users to the view, and verify the users that were added:

```
!Initialize the LI-View.
Device(config)# li-view lipass user li_admin password li_adminpass
Device(config)# end
! Enter the LI-View; that is, check to see what commands are available within the view.
Device# enable view li-view
Password:
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# parser view li-view

Device(config-view)# ?
View commands:
  commands  Configure commands for a view
  default   Set a command to its defaults
  exit      Exit from view configuration mode
  name      New LI-View name
  no        Negate a command or set its defaults
  password  Set a password associated with CLI views
Device(config-view)#
! NOTE:LI View configurations are never shown as part of 'running-configuration'.
! Configure LI Users.
Device(config)# username lawful-intercept li-user1 password li-user1pass

Device(config)# username lawful-intercept li-user2 password li-user2pass
! Displaying LI User information.
Device# show users lawful-intercept
li_admin
li-user1
li-user2
Device#
```

Note The lawful intercept view is available only on specific images and the view name option is available only in the LI view.

Example: Configuring a Superview

The following sample output from the **show running-config** command shows that "view_one" and "view_two" have been added to superview "su_view1", "view_three", and "view_four" have been added to superview "su_view2":

```
Device# show running-config
!
parser view su_view1 superview
 secret 5 <encoded password>
 view view_one
 view view_two
!
parser view su_view2 superview
 secret 5 <encoded password>
 view view_three
```

```
 view view_four
!
```

Additional References for Role-Based CLI Access

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
SNMP, MIBs, CLI configuration	*Cisco IOS Network Management Configuration Guide* , Release 15.0.
Privilege levels	"Configuring Security with Passwords, Privileges and Logins" module.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Role-Based CLI Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 225: Feature Information for Role-Based CLI Access

Feature Name	Releases	Feature Information
Role-Based CLI Access		The Role-Based CLI Access feature enables network administrators to restrict user access to CLI and configuration information. The CLI view capability was extended to restrict user access on a per-interface level, and additional CLI views were introduced to support the extended view capability. Also, support to group configured CLI views into a superview was introduced. The following commands were introduced or modified: **commands (view)**, **enable**, **li-view**, **name (view)**, **parser view , parser view superview**, **secret**, **show parser view**, **show users**, **username**, and **view**.
Role-Based CLI Inclusive Views		The Role-Based CLI Inclusive Views feature enables a standard CLI view including all commands by default. The following command was modified: **parser view inclusive**.

CHAPTER 171

Information About Secure Storage

Secure Storage feature allows you to secure critical configuration information by encrypting it. It encrypts VPN, IPSec, and other asymmetric key-pairs, pre-shared secrets, the type 6 password encryption key and certain credentials. An instance-unique encryption key is stored in the hardware trust anchor to prevent it from being compromised.

By default, this feature is enabled on platforms that come with a hardware trust anchor. This feature is not supported on platforms that do not have hardware trust anchor.

Supported Platforms

Starting from Cisco IOS Release 15.6(3) M1, the following Cisco 880 Series platforms support Secure Storage:

Table 226: Secure Storage Supported Platforms - Cisco Integrated Services Router 880 PID

C881-K9
C887VA-K9
C886VA-K9
C887VAM-K9
C886VAJ-K9
C888-K9

Starting from Cisco IOS Release 15.6(3) M1, the following Cisco 890 Series platforms support Secure Storage:

Table 227: Secure Storage Supported Platforms - Cisco Integrated Services Router 890 PID

C891FW-E-K9

C891F-K9
C891FW-A-K9
C891-24X-K9

Starting from Cisco IOS Release 15.6(3) M1, the following Cisco 800M Series platforms support Secure Storage:

Table 228: Secure Storage Supported Platforms - Cisco Integrated Services Router 800M PID

C841M-4X/K9
C886VA-K9
C841M-8X/K9

Starting from Cisco IOS XE Release 16.6.1, the following ISR 4000 platforms support Secure Storage:

Table 229: Secure Storage Supported Platforms - Cisco Integrated Services Router 4000 PID

ISR4431
ISR4221
ISR4321
ISR4331
ISR4351
ISR4451-X

Starting from Cisco IOS XE Release 16.6.1, the following ASR 1000 platforms support Secure Storage::

Table 230: Secure Storage Supported Platforms - Cisco ASR 1000 Series Aggregation Services Routers PID

ASR1000-RP3
ASR1001-X
ASR1001-HX
ASR1002-HX

Starting from Cisco IOS XE Release 16.9.1, the following Cisco 1000 Series platforms support Secure Storage::

Table 231: Secure Storage Supported Platforms - Cisco 1000 Series PID

C1101-4P
C1111-8P
C1111-4P

C1112-8P
C1113-8P
C1113-8PM
C1116-4P
C1117-4P
C1117-4PM
C1101-4PLTEP
C1111-8PLTEEA
C1111-8PLTELA
C1111-4PLTEEA
C1111-4PLTELA
C1112-8PLTEEA
C1113-8PLTEEA
C1113-8PLTELA
C1113-8PMLTEEA
C1116-4PLTEEA
C1117-4PLTEEA
C1117-4PLTELA
C1117-4PMLTEEA
C1111-8PWY
C1111-4PWX
C1112-8PWE
C1113-8PWA
C1113-8PWB
C1113-8PWE
C1116-4PWE
C1117-4PWE
C1117-4PWA
C1117-4PWZ

C1117-4PMWE
C1111-8PLTEEAWX
C1111-8PLTELAWY
C1112-8PLTEAWE
C1113-8PLTEEAWA
C1113-8PLTEEAWB
C1113-8PLTEEAWE
C1113-8PLTELAWZ
C1116-4PLTEEAWE
C1117-4PMLTEEA
C1117-4PLTEEAWE
C1117-4PLTEEAWA
C1117-4PLTELAWZ
C1117-4PMLTEEAWE
C1101-4PLTEPWX

Enabling Secure Storage

Before you begin

By default, this feature is enabled on a platform. Use this procedure on a platform where it is disabled.

SUMMARY STEPS

1. Config terminal
2. service private-config-encryption
3. do write memory

DETAILED STEPS

	Command or Action	Purpose
Step 1	Config terminal **Example:** `router#config terminal`	Enters the configuration mode.

	Command or Action	Purpose
Step 2	service private-config-encryption **Example:** `router(config)# service private-config-encryption`	Enables the Secure Storage feature on your platform.
Step 3	do write memory **Example:** `router(config)# do write memory`	Encrypts the private-config file and saves the file in an encrypted format.

Example

The following example shows how to enable Secure Storage:

```
router#config terminal
router(config)# service private-config-encryption
router(config)# do write memory
```

Disabling Secure Storage

Before you begin

To disable Secure Storage feature on a platform, perform this task:

SUMMARY STEPS

1. Config terminal
2. no service private-config-encryption
3. do write memory

DETAILED STEPS

	Command or Action	Purpose
Step 1	Config terminal **Example:** `router#config terminal`	Enters the configuration mode.
Step 2	no service private-config-encryption **Example:** `router(config)# no service private-config-encryption`	Disables the Secure Storage feature on your platform.
Step 3	do write memory **Example:** `router(config)# do write memory`	Decrypts the private-config file and saves the file in plane format.

Example

The following example shows how to disable Secure Storage:

```
router#config terminal
router(config)# no service private-config-encryption
router(config)# do write memory
```

Verifying the Status of Encryption

Use the **show parser encrypt file status** command to verify the status of encryption. The following command output indicates that the feature is available but the file is not encrypted. The file is in 'plain text' format.

```
router#show parser encrypt file status
Feature: Enabled
File Format: Plain Text
Encryption Version: Ver1
```

The following command output indicates that the feature is enabled and the file is encrypted. The file is in 'cipher text' format.

```
router#show parser encrypt file status
Feature: Enabled
File Format: Cipher Text
Encryption Version: Ver1
```

Downgrading the Platform Image to an Older Version

Before you downgrade the platform image to an older version where the Secure Storage is not supported, you have to disable the feature in the version where it is supported.

If you do not disable this feature before downgrading to an older image, the private-config file will be in encrypted format. The following Syslog message will be generated to indicate that the file is in encrypted format:

```
%PARSER-4-BADCFG: Unexpected end of configuration file.
```

If the file is in 'plain text', no Syslog message will be generated.

Feature Information for Overview of Secure Storage

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 232: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
Secure Storage	Cisco IOS XE Fuji 16.9.1	The support for Secure Storage is introduced for ASR and ISR platforms.

CHAPTER 172

AutoSecure

The AutoSecure feature secures a router by using a single CLI command to disable common IP services that can be exploited for network attacks, enable IP services and features that can aid in the defense of a network when under attack, and simplify and harden the security configuration of the router.

AutoSecure enhances secure access to the router by configuring a required minimum password length to eliminate common passwords that can be common on many networks, such as “lab” and “company name.” Syslog messages are generated after the number of unsuccessful attempts exceeds the configured threshold.

AutoSecure also allows a router to revert (roll) back to its pre-AutoSecure configuration state if the AutoSecure configuration fails.

When AutoSecure is enabled, a detailed audit trail of system logging messages capture any changes or tampering of the AutoSecure configuration that may have been applied to the running configuration.

Restrictions for AutoSecure

The AutoSecure configuration can be configured at run time or setup time. If any related configuration is modified after AutoSecure has been enabled, the AutoSecure configuration may not be fully effective.

Information About AutoSecure

Securing the Management Plane

The management plane is secured by turning off certain global and interface services that can be potentially exploited for security attacks and turning on global services that help mitigate the threat of attacks. Secure access and secure logging are also configured for the router.

Caution If your device is managed by a network management (NM) application, securing the management plane could turn off some services like the HTTP server and disrupt the NM application support.

The following subsections define how AutoSecure helps to secure the management plane:

Disabling Global Services

After enabling this feature (through the **auto secure** command), the following global services are disabled on the router without prompting the user:

- Finger--Collects information about the system (reconnaissance) before an attack. If enabled, the information can leave your device vulnerable to attacks.
- PAD--Enables all packet assembler and disassembler (PAD) commands and connections between PAD devices and access servers. If enabled, it can leave your device vulnerable to attacks.
- Small Servers--Causes TCP and User Datagram Protocol (UDP) diagnostic port attacks: a sender transmits a volume of fake requests for UDP diagnostic services on the router, consuming all CPU resources.
- Bootp Server--Bootp is an insecure protocol that can be exploited for an attack.
- HTTP Server--Without secure-http or authentication embedded in the HTTP server with an associated ACL, the HTTP server is insecure and can be exploited for an attack. (If you must enable the HTTP server, you are prompted for the proper authentication or access list.)

Note If you are using Cisco Configuration Professional (CCP), you must manually enable the HTTP server through the **ip http server** command.

- Identification Service--An insecure protocol, defined in RFC 1413, that allows one to query a TCP port for identification. An attacker can access private information about the user from the ID server.
- CDP--If a large number of Cisco Discovery Protocol (CDP) packets are sent to the router, the available memory of the router can be consumed, causing the router to crash.

Caution NM applications that use CDP to discover network topology are not able to perform discovery.

- NTP--Without authentication or access-control, Network Time Protocol (NTP) is insecure and can be used by an attacker to send NTP packets to crash or overload the router. (If you want to turn on NTP, you must configure NTP authentication using Message Digest 5 (MD5) and the **ntp access-group** command. If NTP is enabled globally, disable it on all interfaces on which it is not needed.)
- Source Routing--Provided only for debugging purposes, so source routing should be disabled in all other cases. Otherwise, packets may slip away from some of the access control mechanisms that they should have gone through.

Disabling Per Interface Services

After enabling this feature, the following per interface services are disabled on the router without prompting the user:

- ICMP redirects--Disabled on all interfaces. Does not add a useful functionality to a correctly configured to network, but it could be used by attackers to exploit security holes.
- ICMP unreachables--Disabled on all interfaces. Internet Control Management Protocol (ICMP) unreachables are a known cause for some ICMP-based denial of service (DoS) attacks.
- ICMP mask reply messages--Disabled on all interfaces. ICMP mask reply messages can give an attacker the subnet mask for a particular subnetwork in the internetwork.
- Proxy-Arp--Disabled on all interfaces. Proxy-Arp requests are a known cause for DoS attacks because the available bandwidth and resources of the router can be consumed in an attempt to respond to the repeated requests that are sent by an attacker.
- Directed Broadcast--Disabled on all interfaces. Potential cause of SMURF attacks for DoS.
- Maintenance Operations Protocol (MOP) service--Disabled on all interfaces.

Enabling Global Services

After AutoSecure is enabled, the following global services are enabled on the router without prompting the user:

- The **service password-encryption** command--Prevents passwords from being visible in the configuration.
- The **service tcp-keepalives-in** and **service tcp-keepalives-out** commands--Ensures that abnormally terminated TCP sessions are removed.

Securing Access to the Router

Caution If your device is managed by an NM application, securing access to the router could turn off vital services and may disrupt the NM application support.

After enabling this feature, the following options in which to secure access to the router are available to the user:

- If a text banner does not exist, users are prompted to add a banner. This feature provides the following sample banner:

```
Authorized access only
This system is the property of ABC Enterprise
Disconnect IMMEDIATELY if you are not an authorized user!
Contact abc@xyz.com +99 876 543210 for help.
```

- The login and password (preferably a secret password, if supported) are configured on the console, AUX, vty, and tty lines. The **transport input** and **transport output**commandsare also configured on all of these lines. (Telnet and secure shell (SSH) are the only valid transport methods.) The **exec-timeout** command is configured on the console and AUX as 10.

- When the image on the device is a crypto image, AutoSecure enables SSH and secure copy (SCP) for access and file transfer to and from the router. The **timeout** *seconds* and **authentication-retries** *integer* options for the **ip ssh** command are configured to a minimum number. (Telnet and FTP are not affected by this operation and remain operational.)
- If the AutoSecure user specifies that their device does not use Simple Network Management Protocol (SNMP), one of the following functions occur:
  - In interactive mode, the user is asked whether to disable SNMP regardless of the values of the community strings, which act like passwords to regulate access to the agent on the router.
  - In non-interact mode, SNMP is disabled if the community string is “public” or “private.”

Note After AutoSecure has been enabled, tools that use SNMP to monitor or configure a device is unable to communicate with the device through SNMP.

- If authentication, authorization, and accounting (AAA) is not configured, configure local AAA. AutoSecure prompts users to configure a local username and password on the router.

Security Logging

The following logging options are available after AutoSecure is enabled. These options identify security incidents and provide ways to respond to them.

- Sequence numbers and time stamps for all debug and log messages. This option is useful when auditing logging messages.
- Logging messages can be generated for login-related events; for example, the message “Blocking Period when Login Attack Detected” is displayed when a login attack is detected and the router enters “quiet mode.” (Quiet mode means that the router does not allow any login attempts through Telnet, HTTP, or SSH.)

For more information on login system messages, see the Cisco IOS Release 12.3(4)T feature module Cisco IOS Login Enhancements .

- The **logging console critical**command, which sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
- The **logging buffered** command, which copies logging messages to an internal buffer and limits messages logged to the buffer based on severity.
- The **logging trap debugging** command, which allows all commands with a severity higher than debugging to be sent to the logging server.

Securing the Forwarding Plane

To minimize the risk of attacks on the router forward plane, AutoSecure provides the following functions:

- Cisco Express Forwarding (CEF)--AutoSecure enables CEF or distributed CEF (dCEF) on the router whenever possible. Because there is no need to build cache entries when traffic starts arriving for new destinations, CEF behaves more predictably than other modes when presented with large volumes of

traffic addressed to many destinations. Thus, routers configured for CEF perform better under SYN attacks than routers using the traditional cache.

Note CEF consumes more memory than a traditional cache.

- If the TCP intercept feature is available, it can be configured on the router for connection timeout.
- If strict Unicast Reverse Path Forwarding (uRPF) is available, it can be configured on the router to help mitigate problems that are caused by the introduction of forged (spoofed) IP source addresses. uRPF discards IP packets that lack a verifiable IP source address.
- If the router is being used as a firewall, it can be configured for context-based access control (CBAC) on public interfaces that are facing the Internet.

Note At the beginning of the AutoSecure dialogue, you are prompted for a list of public interfaces.

How to Configure AutoSecure

Configuring AutoSecure

Caution Although the **auto secure**command helps to secure a router, it does not guarantee the complete security of the router.

SUMMARY STEPS

1. **enable**
2. auto secure [management | forwarding] [no-interact | full] [ntp | login | ssh | firewall | tcp-intercept]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	auto secure [management | forwarding] [no-interact | full] [ntp | login | ssh | firewall | tcp-intercept] **Example:**	A semi-interactive dialogue session begins to secure either the management or forwarding planes on the router when the **management** or **forwarding** keyword is selected. If neither option is selected, then the dialogue asks for both

	Command or Action	Purpose
	`Router# auto secure`	planes to be configured. If the **management** keyword is selected, then the management plane is secured only. If the **forwarding keyword is selected, then**the forwarding plane is secured only. If the **no-interact** keyword is selected, then the user is not prompted for any interactive configurations. If the **full** keyword is selected, then user is prompted for all interactive questions, which is the default.

Configuring Enhanced Security Access to the Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable password** {*password* | [*encryption-type*] *encrypted-password* }
4. **security authentication failure rate** *threshold-rate* **log**
5. **exit** *threshold-rate* **log**
6. **show auto secure config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**enable password** {*password* | [*encryption-type*] *encrypted-password* } **Example:** `Router(config)# enable password elephant`	Sets a local password to control access to various privilege levels.
Step 4	**security authentication failure rate** *threshold-rate* **log** **Example:** `Router(config)# security authentication failure rate 10 log`	Configures the number of allowable unsuccessful login attempts. • *threshold-rate* --Number of allowable unsuccessful login attempts.

	Command or Action	Purpose
		• **log** --Syslog authentication failures if the rate exceeds the threshold.
Step 5	**exit** *threshold-rate* **log** **Example:** `Router(config)# exit`	Exits configuration mode and enters privileged EXEC mode.
Step 6	**show auto secure config** **Example:** `Router# show auto secure config`	(Optional) Displays all configuration commands that have been added as part of the AutoSecure configuration.

Configuration Example for AutoSecure

The following example is a sample AutoSecure dialogue. After you enable the **auto secure** command, the feature automatically prompts you with a similar dialogue unless you enable the **no-interact** keyword. (For information on which services are disabled and which features are enabled, see the sections, Securing the Management Plane, on page 2253 and Securing the Forwarding Plane, on page 2256 earlier in this document.)

```
Router# auto secure
               --- AutoSecure Configuration ---
*** AutoSecure configuration enhances the security of the router but it will not make router
 absolutely secure from all security attacks ***
All the configuration done as part of AutoSecure will be shown here. For more details of
why and how this configuration is useful, and any possible side effects, please refer to
Cisco documentation of AutoSecure.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.
Gathering information about the router for AutoSecure
Is this router connected to internet? [no]:y
Enter the number of interfaces facing internet [1]:
Interface                  IP-Address OK? Method Status
Protocol
FastEthernet0/1/0          10.1.1.1  YES NVRAM  up down
FastEthernet1/0/0            10.2.2.2  YES NVRAM  up down
FastEthernet1/1/0            10.0.0.1  YES NVRAM  up up
Loopback0                  unassigned YES NVRAM  up up
FastEthernet0/0/0            10.0.0.2  YES NVRAM  up down
Enter the interface name that is facing internet:FastEthernet0/0/0
Securing Management plane services..
Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol
Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp
```

```
Enable secret is either not configured or is same as enable password
Enter the new enable secret:abc123
Configuring aaa local authentication
Configuring console, Aux and vty lines for
local authentication, exec-timeout, transport
Configure SSH server? [yes]:
Enter the domain-name:example.com
Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
Disabling mop on Ethernet interfaces
Securing Forwarding plane services..
Enabling CEF (it might have more memory requirements on some low end
platforms)
Enabling unicast rpf on all interfaces connected to internet
Configure CBAC Firewall feature? [yes/no]:yes
This is the configuration generated:
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
security authentication failure rate 10 log
enable secret 5 $1$CZ6G$GkGOnHdNJCO3CjNHHyTUA.
aaa new-model
aaa authentication login local_auth local
line console 0
 login authentication local_auth
 exec-timeout 5 0
 transport output telnet
line aux 0
 login authentication local_auth
 exec-timeout 10 0
 transport output telnet
line vty 0 4
 login authentication local_auth
 transport input telnet
ip domain-name example.com
crypto key generate rsa general-keys modulus 1024
ip ssh time-out 60
ip ssh authentication-retries 2
line vty 0 4
 transport input ssh telnet
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
interface FastEthernet0/1/0
```

```
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface FastEthernet1/0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface FastEthernet1/1/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface FastEthernet0/0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
ip cef
interface FastEthernet0/0/0
 ip verify unicast reverse-path
ip inspect audit-trail
ip inspect dns-timeout 7
ip inspect tcp idle-time 14400
ip inspect udp idle-time 1800
ip inspect name autosec_inspect cuseeme timeout 3600
ip inspect name autosec_inspect ftp timeout 3600
ip inspect name autosec_inspect http timeout 3600
ip inspect name autosec_inspect rcmd timeout 3600
ip inspect name autosec_inspect realaudio timeout 3600
ip inspect name autosec_inspect smtp timeout 3600
ip inspect name autosec_inspect tftp timeout 30
ip inspect name autosec_inspect udp timeout 15
ip inspect name autosec_inspect tcp timeout 3600
access-list 100 deny ip any any
interface FastEthernet0/0/0
 ip inspect autosec_inspect out
 ip access-group 100 in
!
end
Apply this configuration to running-config? [yes]:yes
Applying the config generated to running-config
The name for the keys will be:ios210.example.com
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys ...[OK]
Router#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring SNMP Support	Configuring SNMP Support
Security Commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
PacketCable™ *Control Point Discovery Interface Specification*	*PacketCable™ Control Point Discovery Interface Specification* (PKT-SP-CPD-I02-061013)

MIBs

MIB	MIBs Link
• CISCO-802-TAP-MIB • CISCO-IP-TAP-MIB • CISCO-MOBILITY-TAP-MIB • CISCO-TAP2-MIB • CISCO-USER-CONNECTION-TAP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC-2865	*Remote Authentication Dial In User Service (RADIUS)*
RFC-3576	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*
RFC-3924	*Cisco Architecture for Lawful Intercept in IP Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for AutoSecure

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 233: Feature Information for AutoSecure

Feature Name	Releases	Feature Information
AutoSecure Manageability	Cisco IOS XE Release 2.3	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. By using a single command-line interface (CLI), the AutoSecure feature allows a user to perform the following functions: • Disable common IP services that can be exploited for network attacks • Enable IP services and features that can aid in the defense of a network when under attack This feature also simplifies the security configuration of a router and hardens the router configuration. The following commands were introduced or modified: **auto secure** and **show auto secure config**

CHAPTER 173

Configuring Kerberos

Information About Kerberos

Kerberos is a secret-key network authentication protocol, developed at the Massachusetts Institute of Technology (MIT), that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos, like other secret-key systems, is based on the concept of a trusted third party that performs secure verification of users and services. In the Kerberos protocol, this trusted third party is called the key distribution center (KDC).

The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

The Kerberos credential scheme embodies a concept called "single logon." This process requires authenticating a user once, and then allows secure authentication (without encrypting another password) wherever that user's credential is accepted.

Cisco IOS XE software includes Kerberos 5 support, which allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs).

The following network services are supported by the Kerberos authentication capabilities in Cisco IOS XE software:

- Telnet
- rlogin
- rsh
- rcp

Note Cisco Systems' implementation of Kerberos client support is based on code developed by CyberSafe, which was derived from the MIT code. As a result, the Cisco Kerberos implementation has successfully undergone full compatibility testing with the CyberSafe Challenger commercial Kerberos server and MIT's server code, which is freely distributed.

The table below lists common Kerberos-related terms and their definitions.

Table 234: Kerberos Terminology

Term	Definition
authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a router or a router can authenticate to another router.
authorization	A means by which the router determines what privileges you have in a network or on the router and what actions you can perform.
credential	A general term that refers to authentication tickets, such as ticket granting tickets (TGTs) and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued a ticket, it can be used in place of retyping in a username and password. Credentials have a default lifespan of eight hours.
instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. It is up to the server of each network service to implement and enforce the authorization mappings of Kerberos instances. Note that the Kerberos realm name must be in uppercase characters.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
key distribution center (KDC)	A Kerberos server and database program running on a network host.
principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server.

Term	Definition
service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC, and with the user's TGT.
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
ticket granting ticket (TGT)	A credential that the key distribution center (KDC) issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

Kerberos Client Support Operation

This section describes how the Kerberos security system works with a Cisco router functioning as the security server. Although (for convenience or technical reasons) you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

Authenticating to the Boundary Router

This section describes the first layer of security that remote users must pass through when they attempt to access a network. The first step in the Kerberos authentication process is for users to authenticate themselves to the boundary router. The following process describes how users authenticate to a boundary router:

1. The remote user opens a PPP connection to the corporate site router.
2. The router prompts the user for a username and password.
3. The router requests a TGT from the KDC for this particular user.
4. The KDC sends an encrypted TGT to the router that includes (among other things) the user's identity.
5. The router attempts to decrypt the TGT using the password the user entered. If the decryption is successful, the remote user is authenticated to the router.

A remote user who successfully initiates a PPP session and authenticates to the boundary router is inside the firewall but still must authenticate to the KDC directly before being allowed to access network services. This is because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on to the router.

Obtaining a TGT from a KDC

This section describes how remote users who are authenticated to the boundary router authenticate themselves to a KDC.

When a remote user authenticates to a boundary router, that user technically becomes part of the network; that is, the network is extended to include the remote user and the user's machine or network. To gain access to network services, however, the remote user must obtain a TGT from the KDC. The following process describes how remote users authenticate to the KDC:

1. The remote user, at a workstation on a remote site, launches the KINIT program (part of the client software provided with the Kerberos protocol).
2. The KINIT program finds the user's identity and requests a TGT from the KDC.
3. The KDC creates a TGT, which contains the identity of the user, the identity of the KDC, and the expiration time of the TGT.
4. Using the user's password as a key, the KDC encrypts the TGT and sends the TGT to the workstation.
5. When the KINIT program receives the encrypted TGT, it prompts the user for a password (this is the password that is defined for the user in the KDC).
6. If the KINIT program can decrypt the TGT with the password the user enters, the user is authenticated to the KDC, and the KINIT program stores the TGT in the user's credential cache.

At this point, the user has a TGT and can communicate securely with the KDC.In turn, the TGT allows the user to authenticate to other network services.

Authenticating to Network Services

The following process describes how a remote user with a TGT authenticates to network services within a given Kerberos realm. Assume the user is on a remote workstation (Host A) and wants to log in to Host B.

1. The user on Host A initiates a Kerberized application (such as Telnet) to Host B.
2. The Kerberized application builds a service credential request and sends it to the KDC. The service credential request includes (among other things) the user's identity and the identity of the desired network service. The TGT is used to encrypt the service credential request.
3. The KDC tries to decrypt the service credential request with the TGT it issued to the user on Host A. If the KDC can decrypt the packet, it is assured that the authenticated user on Host A sent the request.
4. The KDC notes the network service identity in the service credential request.
5. The KDC builds a service credential for the appropriate network service on Host B on behalf of the user on Host A. The service credential contains the client's identity and the desired network service's identity.
6. The KDC then encrypts the service credential twice. It first encrypts the credential with the SRVTAB that it shares with the network service identified in the credential. It then encrypts the resulting packet with the TGT of the user (who, in this case, is on Host A).
7. The KDC sends the twice-encrypted credential to Host A.
8. Host A attempts to decrypt the service credential with the user's TGT. If Host A can decrypt the service credential, it is assured the credential came from the real KDC.
9. Host A sends the service credential to the desired network service. Note that the credential is still encrypted with the SRVTAB shared by the KDC and the network service.
10. The network service attempts to decrypt the service credential using its SRVTAB.
11. If the network service can decrypt the credential, it is assured the credential was in fact issued from the KDC. Note that the network service trusts anything it can decrypt from the KDC, even if it receives it indirectly from a user. This is because the user first authenticated with the KDC.

At this point, the user is authenticated to the network service on Host B. This process is repeated each time a user wants to access a network service in the Kerberos realm.

How to Configure Kerberos

For hosts and the KDC in your Kerberos realm to communicate and mutually authenticate, you must identify them to each other. To do this, you add entries for the hosts to the Kerberos database on the KDC and add SRVTAB files generated by the KDC to all hosts in the Kerberos realm. You also make entries for users in the KDC database.

This section describes how to set up a Kerberos-authenticated server-client system and contains the following topics:

This section assumes that you have installed the Kerberos administrative programs on a UNIX host, known as the KDC, initialized the database, and selected a Kerberos realm name and password. For instructions about completing these tasks, refer to documentation that came with your Kerberos software.

Note Write down the host name or IP address of the KDC, the port number you want the KDC to monitor for queries, and the name of the Kerberos realm it will serve. You need this information to configure the router.

Configuring the KDC Using Kerberos Commands

After you set up a host to function as the KDC in your Kerberos realm, you must make entries to the KDC database for all principals in the realm. Principals can be network services on Cisco routers and hosts or they can be users.

To use Kerberos commands to add services to the KDC database (and to modify existing database information), complete the tasks in the following sections:

Note All Kerberos command examples are based on Kerberos 5 Beta 5 of the original MIT implementation. Later versions use a slightly different interface.

Adding Users to the KDC Database

To add users to the KDC and create privileged instances of those users, use the **su** command to become root on the host running the KDC and use the kdb5_edit program to use the following commands in privileged EXEC mode:

SUMMARY STEPS

1. Router# **ank***username@REALM*
2. Router# **ank***username/instance@REALM*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# **ank***username@REALM*	Use the **ank** (add new key) command to add a user to the KDC. This command prompts for a password, which the user must enter to authenticate to the router.
Step 2	Router# **ank***username/instance@REALM*	Use the **ank** command to add a privileged instance of a user.

What to do next

For example, to add user *loki* of Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ank loki@CISCO.COM
```

Note The Kerberos realm name must be in uppercase characters.

You might want to create privileged instances to allow network administrators to connect to the router at the enable level, for example, so that they need not enter a clear text password (and compromise security) to enter enable mode.

To add an instance of *loki* with additional privileges (in this case, enable, although it could be anything) enter the following Kerberos command:

```
ank loki/enable@CISCO.COM
```

In each of these examples, you are prompted to enter a password, which you must give to user *loki* to use at login.

The Enabling Kerberos Instance Mapping, on page 2275 describes how to map Kerberos instances to various Cisco IOS XE privilege levels.

Creating SRVTABs on the KDC

All routers that you want to authenticate to use the Kerberos protocol must have an SRVTAB. For more information on extracting SRVTABs, see *Extracting SRVTABs*

To make SRVTAB entries on the KDC, use the following command in privileged EXEC mode:

Command	Purpose
Router# **ark** *SERVICE/HOSTNAME@REALM*	Use the **ark** (add random key) command to add a network service supported by a host or router to the KDC.

For example, to add a Kerberized authentication service for a Cisco router called *router1* to the Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ark host/router1.cisco.com@CISCO.COM
```

Make entries for all network services on all Kerberized hosts that use this KDC for authentication.

Extracting SRVTABs

SRVTABs contain (among other things) the passwords or randomly generated keys for the service principals you entered into the KDC database. Service principal keys must be shared with the host running that service. To do this, you must save the SRVTAB entries to a file, then copy the file to the router and all hosts in the Kerberos realm. Saving SRVTAB entries to a file is called *extracting* SRVTABs. To extract SRVTABs, use the following command in privileged EXEC mode:

Command	Purpose
`Router# xst router-name host`	Use the kdb5_edit command **xst** to write an SRVTAB entry to a file.

For example, to write the host/router1.cisco.com@CISCO.COM SRVTAB to a file, enter the following Kerberos command:

```
xst router1.cisco.com@CISCO.COM host
```

Use the **quit** command to exit the kdb5_edit program.

Configuring the Router to Use the Kerberos Protocol

Defining a Kerberos Realm

For a router to authenticate a user defined in the Kerberos database, it must know the host name or IP address of the host running the KDC, the name of the Kerberos realm and, optionally, be able to map the host name or Domain Name System (DNS) domain to the Kerberos realm.

To configure the router to authenticate to a specified KDC in a specified Kerberos realm, use the following commands in global configuration mode. Note that DNS domain names must begin with a leading dot (.):

SUMMARY STEPS

1. Router(config)# **kerberos local-realm***kerberos-realm*
2. Router(config)# **kerberos server***kerberos-realm* {*hostname* | *ip-address* } [*port-number*]
3. Router(config)# **kerberos realm** {*dns-domain* | *host* } *kerberos-realm*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **kerberos local-realm***kerberos-realm*	Defines the default realm for the router.
Step 2	Router(config)# **kerberos server***kerberos-realm* {*hostname* | *ip-address* } [*port-number*]	Specifies to the router which KDC to use in a given Kerberos realm and, optionally, the port number that the KDC is monitoring. (The default is 88.)
Step 3	Router(config)# **kerberos realm** {*dns-domain* | *host* } *kerberos-realm*	(Optional) Maps a host name or DNS domain to a Kerberos realm.

What to do next

Note Because the machine running the KDC and all Kerberized hosts must interact within a 5-minute window or authentication fails, all Kerberized machines, and especially the KDC, should be running the Network Time Protocol (NTP).

The **kerberos local-realm**, **kerberos realm**, and **kerberos server** commands are equivalent to the UNIX krb.conf file. The table below identifies mappings from the Cisco IOS XE configuration commands to a Kerberos 5 configuration file (krb5.conf).

Table 235: Kerberos 5 Configuration File and Commands

krb5.conf File	Cisco IOS XE Configuration Command
[libdefaults] `default_realm = DOMAIN.COM`	(in configuration mode) **kerberos local-realm** *DOMAIN.COM*

[domain_realm]
 	
`.domain.com = DOMAIN.COM`
 	
`domain.com = DOMAIN.COM`
	(in configuration mode)
	
	kerberos realm
	.domain.com
	DOMAIN.COM
	kerberos realm
	domain.com DOMAIN.COM

[realms]
 	
`kdc = DOMAIN.PIL.COM:750`
 	
`admin_server = DOMAIN.PIL.COM`
 	
`default_domain = DOMAIN.COM`
	(in configuration mode)
	
	kerberos server
	DOMAIN.COM 172.65.44.2
	(172.65.44.2
	is the example IP address for *DOMAIN.PIL.COM*
	)

For an example of defining a Kerberos realm, see the Defining a Kerberos Realm Examples module.

Copying SRVTAB Files

To make it possible for remote users to authenticate to the router using Kerberos credentials, the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and go to each host in turn and manually copy the files onto the system. To copy SRVTAB files to the router, which does not have a physical media drive, you must transfer them via the network using TFTP.

To remotely copy SRVTAB files to the router from the KDC, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos srvtab remote {hostname | ip-address} {filename}`	Retrieves an SRVTAB file from the KDC.

When you copy the SRVTAB file from the router to the KDC, the **kerberos srvtab remote** command parses the information in this file and stores it in the router's running configuration in the **kerberos srvtab entry**format. To ensure that the SRVTAB is available (does not need to be acquired from the KDC) when you reboot the router, use the **write memory** configuration command to write your running configuration (which contains the parsed SRVTAB file) to NVRAM.

For an example of copying SRVTAB files, see the SRVTAB File Copying Example, on page 2276.

Specifying Kerberos Authentication

You have now configured Kerberos on your router. This makes it possible for the router to authenticate using Kerberos. The next step is to tell it to do so. Because Kerberos authentication is facilitated through AAA, you need to enter the **aaa authentication** command, specifying Kerberos as the authentication method. For more information, refer to the chapter "Configuring Authentication".

Enabling Credentials Forwarding

With Kerberos configured thus far, a user authenticated to a Kerberized router has a TGT and can use it to authenticate to a host on the network. However, if the user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

You can optionally configure the router to forward users' TGTs with them as they authenticate from the router to Kerberized remote hosts on the network when using Kerberized Telnet, rcp, rsh, and rlogin (with the appropriate flags).

To force all clients to forward users' credentials as they connect to other hosts in the Kerberos realm, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos credentials forward`	Forces all clients to forward user credentials upon successful Kerberos authentication.

With credentials forwarding enabled, users' TGTs are automatically forwarded to the next host they authenticate to. In this way, users can connect to multiple hosts in the Kerberos realm without running the KINIT program each time to get a new TGT.

Opening a Telnet Session to the Router

To use Kerberos to authenticate users opening a Telnet session to the router from within the network, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication login {default | list-name } krb5_telnet`	Sets login authentication to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router.

Although Telnet sessions to the router are authenticated, users must still enter a clear text password if they want to enter enable mode. The **kerberos instance map** command, discussed in a later section, allows them to authenticate to the router at a predefined privilege level.

Establishing an Encrypted Kerberized Telnet Session

Another way for users to open a secure Telnet session is to use Encrypted Kerberized Telnet. With Encrypted Kerberized Telnet, users are authenticated by their Kerberos credentials before a Telnet session is established. The Telnet session is encrypted using 56-bit Data Encryption Standard (DES) encryption with 64-bit Cipher Feedback (CFB). Because data sent or received is encrypted, not clear text, the integrity of the dialed router or access server can be more easily controlled.

Note This feature is available only if you have the 56-bit encryption image. 56-bit DES encryption is subject to U.S. Government export control regulations.

To establish an encrypted Kerberized Telnet session from a router to a remote host, use either of the following commands in EXEC command mode:

Command	Purpose

`Router(config)# connect host [port ] /encrypt kerberos`
 	
or
 	
`Router(config)# telnet host [port ] /encrypt kerberos`
	Establishes an encrypted Telnet session.

When a user opens a Telnet session from a Cisco router to a remote host, the router and remote host negotiate to authenticate the user using Kerberos credentials. If this authentication is successful, the router and remote host then negotiate whether or not to use encryption. If this negotiation is successful, both inbound and outbound traffic is encrypted using 56-bit DES encryption with 64-bit CFB.

When a user dials in from a remote host to a Cisco router configured for Kerberos authentication, the host and router will attempt to negotiate whether or not to use encryption for the Telnet session. If this negotiation is successful, the router will encrypt all outbound data during the Telnet session.

If encryption is not successfully negotiated, the session will be terminated and the user will receive a message stating that the encrypted Telnet session was not successfully established.

For information about enabling bidirectional encryption from a remote host, refer to the documentation specific to the remote host device.

For an example of using encrypted Kerberized Telnet to open a secure Telnet session, see the Encrypted Telnet Session Example, on page 2277.

Enabling Mandatory Kerberos Authentication

As an added layer of security, you can optionally configure the router so that, after remote users authenticate to it, these users can authenticate to other services on the network only with Kerberized Telnet, rlogin, rsh, and rcp. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service; for example, Telnet and rlogin prompt for a password, and rsh attempts to authenticate using the local rhost file.

To make Kerberos authentication mandatory, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos clients mandatory`	Sets Telnet, rlogin, rsh, and rcp to fail if they cannot negotiate the Kerberos protocol with the remote server.

Enabling Kerberos Instance Mapping

As mentioned in the Creating SRVTABs on the KDC, on page 2270, you can create administrative instances of users in the KDC database. The **kerberos instance map** command allows you to map those instances to Cisco IOS XE privilege levels so that users can open secure Telnet sessions to the router at a predefined privilege level, obviating the need to enter a clear text password to enter enable mode.

To map a Kerberos instance to a Cisco IOS XE privilege level, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos instance map` *instance privilege-level*	Maps a Kerberos instance to a Cisco IOS XE privilege level.

If there is a Kerberos instance for user *loki* in the KDC database (for example, *loki/admin*), user *loki* can now open a Telnet session to the router as loki/admin and authenticate automatically at privilege level 15, assuming instance “admin” is mapped to privilege level 15. (See theOpening a Telnet Session to the Router, on page 2273.)

Cisco IOS XE commands can be set to various privilege levels using the **privilege level**command.

After you map a Kerberos instance to a Cisco IOS XE privilege level, you must configure the router to check for Kerberos instances each time a user logs in. To run authorization to determine if a user is allowed to run an EXEC shell based on a mapped Kerberos instance, use the **aaa authorization** command with the **krb5-instance** keyword. For more information, refer to the chapter “Configuring Authorization.”

Monitoring and Maintaining Kerberos

To display or remove a current user’s credentials, use the following commands in EXEC mode:

SUMMARY STEPS

1. Router# **show kerberos creds**
2. Router# **clear kerberos creds**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# **show kerberos creds**	Lists the credentials in a current user's credentials cache.
Step 2	Router# **clear kerberos creds**	Destroys all credentials in a current user's credentials cache, including those forwarded.

Kerberos Configuration Examples

Kerberos Realm Definition Examples

To define CISCO.COM as the default Kerberos realm, use the following command:

```
kerberos local-realm CISCO.COM
```

To tell the router that the CISCO.COM KDC is running on host 10.2.3.4 at port number 170, use the following Kerberos command:

```
kerberos server CISCO.COM 10.2.3.4 170
```

To map the DNS domain cisco.com to the Kerberos realm CISCO.COM, use the following command:

```
kerberos realm.cisco.com CISCO.COM
```

SRVTAB File Copying Example

To copy over the SRVTAB file on a host named host123.cisco.com for a router named router1.cisco.com, the command would look like this:

```
kerberos srvtab remote host123.cisco.com router1.cisco.com-new-srvtab
Valid Starting          Expires                 Service Principal
13-May-1996 14:59:44    13-May-1996 23:00:45    krbtgt/CISCO.COM@CISCO.COM
chet-2500# show privilege
Current privilege level is 15
chet-2500# q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.
User Access Verification
Username: chet/restricted
Password:
chet-2500# show kerberos creds
Default Principal:  chet/restricted@CISCO.COM
```

```
Valid Starting          Expires                 Service Principal
13-May-1996 15:00:32    13-May-1996 23:01:33    krbtgt/CISCO.COM@CISCO.COM
chet-2500# show privilege
Current privilege level is 3
chet-2500# q
Connection closed by foreign host.
chet-ss20%
```

Encrypted Telnet Session Example

The following example shows how to establish an encrypted Telnet session from a router to a remote host named “host1”:

Router> **telnet host1 /encrypt kerberos**

Additional References

The following sections provide references related to the No Service Password-Recovery feature.

Related Documents

Related Topic	Document Title
Setting, changing, and recovering lost passwords	“ Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices ” feature module
Loading system images and rebooting	“ Using the Cisco IOS Integrated File System ” feature module
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Kerberos

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 236: Feature Information for Configuring Kerberos

Feature Name	Releases	Feature Information
Encrypted Kerberized Telnet	Cisco IOS XE Release 2.1	With Encrypted Kerberized Telnet, users are authenticated by their Kerberos credentials before a Telnet session is established. The Telnet session is encrypted using 56-bit Data Encryption Standard (DES) encryption with 64-bit Cipher Feedback (CFB). Because data sent or received is encrypted, not clear text, the integrity of the dialed router or access server can be more easily controlled. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **connect**, and **telnet**.

Feature Name	Releases	Feature Information
Kerberos V Client Support	Cisco IOS XE Release 2.1	Kerberos 5 support allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs). In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

CHAPTER 174

Lawful Intercept Architecture

The Lawful Intercept (LI) feature supports service providers in meeting the requirements of law enforcement agencies (LEA) to provide electronic surveillance as authorized by a judicial or administrative order. The surveillance is performed using wiretaps to intercept Voice-over-Internet protocol (VoIP) or data traffic going through the edge routers. The LEA delivers a request for a wiretap to the target's service provider, who is responsible for intercepting data communication to and from the individual using IP sessions.

This document explains LI architecture, including Cisco Service Independent Intercept architecture and PacketCable Lawful Intercept architecture. It also describes the components of the LI feature and provides instructions on how to configure the LI feature in your system.

Before Cisco IOS XE Release 2.5, PPP sessions were tapped based on the accounting session. Circuit-ID based tapping was introduced in Cisco IOS XE Release 2.5.

In Cisco IOS XE Release 2.6, a user session is tapped based on the unique PPP over Ethernet (PPPoE) circuit ID tag. This circuit ID tag serves as a unique parameter for the PPPoE user session on the device. The tapped user session is provisioned through SNMP, and user session data packets and RADIUS authentication data packets are tapped.

Prerequisites for Lawful Intercept

Access to the Cisco LI MIB view should be restricted to the mediation device and to system administrators who need to be aware of lawful intercepts on the router. To access the MIB, users must have level-15 access rights on the router.

Communication with Mediation Device

For the router to communicate with the mediation device to execute a lawful intercept, the following configuration requirements must be met:

- The domain name for both the router and the mediation device must be registered in the Domain Name System (DNS).

In DNS, the router IP address is typically the address of the FastEthernet0/0/0 interface on the router.

- The mediation device must have an access function (AF) and an access function provisioning interface (AFPI).
- You must add the mediation device to the Simple Network Management Protocol (SNMP) user group that has access to the CISCO-TAP2-MIB view. Specify the username of the mediation device as the user to add to the group.

When you add the mediation device as a CISCO-TAP2-MIB user, you can include the mediation device's authorization password if you want. The password must be at least eight characters in length.

Restrictions for Lawful Intercept

General Restrictions

There is no command-line interface (CLI) available to configure LI on the router. All error messages are sent to the mediation device as SNMP notifications. All intercepts are provisioned using SNMPv3 only.

Lawful Intercept does not support SUP HA. LI configuration needs to be reapplied after SUP switchover. An SNMP trap will be generated for this event.

Lawful Intercept MIBs

Only the mediation device and users who need to know about lawful intercepts are allowed to access the LI MIBs.

Due to its sensitive nature, the Cisco LI MIBs are only available in software images that support the LI feature. These MIBs are not accessible through the Network Management Software MIBs Support page (http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml).

SNMP Notifications

SNMP notifications for LI must be sent to User Datagram Protocol (UDP) port 161 on the mediation device, not port 162 (which is the SNMP default).

Information About Lawful Intercept

Introduction to Lawful Intercept

LI is the process by which law enforcement agencies (LEAs) conduct electronic surveillance as authorized by judicial or administrative order. Increasingly, legislation is being adopted and regulations are being enforced that require service providers (SPs) and Internet service providers (ISPs) to implement their networks to explicitly support authorized electronic surveillance. The types of SPs or ISPs that are subject to LI mandates vary greatly from country to country. LI compliance in the United States is specified by the Commission on Accreditation for Law Enforcement Agencies (CALEA).

Cisco supports two architectures for LI: PacketCable and Service Independent Intercept. The LI components by themselves do not ensure customer compliance with applicable regulations but rather provide tools that can be used by SPs and ISPs to construct an LI-compliant network.

Cisco Service Independent Intercept Architecture

The Cisco Service Independent Intercept Architecture Version 3.0 document describes implementation of LI for VoIP networks using the Cisco BTS 10200 Softswitch call agent, version 5.0, in a non-PacketCable network. Packet Cable Event Message specification version 1.5-I01 is used to deliver the call identifying information along with version 2.0 of the Cisco Tap MIB for call content.

The Cisco Service Independent Intercept Architecture Version 2.0 document describes implementation of LI for VoIP networks using the Cisco BTS 10200 Softswitch call agent, versions 4.4 and 4.5, in a non-PacketCable network. Although not a PacketCable network, PacketCable Event Messages Specification version I08 is still used to deliver call identifying information, along with version 1.0 or version 2.0 of the Cisco Tap MIB for call content. The *Cisco Service Independent Intercept Architecture Version 2.0* document adds additional functionality for doing data intercepts by both IP address and session ID, which are both supported in version 2.0 of the Cisco Tap MIB (CISCO-TAP2-MIB).

The Cisco Service Independent Intercept Architecture Version 1.0 document describes implementation of LI for VoIP networks that are using the Cisco BTS 10200 Softswitch call agent, versions 3.5 and 4.1, in a non-PacketCable network. Although not a PacketCable network, PacketCable Event Message Specification version I03 is still used to deliver call identifying information, along with version 1.0 of the Cisco Tap MIB (CISCO-TAP-MIB) for call content. Simple data intercepts by IP address are also discussed.

PacketCable Lawful Intercept Architecture

The P*acketCable Lawful Intercept Architecture for BTS Version 5.0* document describes the implementation of LI for VoIP using Cisco BTS 10200 Softswitch call agent, version 5.0, in a PacketCable network that conforms to PacketCable Event Messages Specification version 1.5-I01.

The *PacketCable Lawful Intercept Architecture for BTS Versions 4.4 and 4.5* document describes the implementation of LI for VoIP using Cisco BTS 10200 Softswitch call agent, versions 4.4 and 4.5, in a PacketCable network that conforms to PacketCable Event Messages Specification version I08.

The PacketCable Lawful Intercept Architecture for BTS Versions 3.5 and 4.1 document describes the implementation of LI for voice over IP (VoIP) using Cisco Broadband Telephony Softswitch (BTS) 10200 Softswitch call agent, versions 3.5 and 4.1, in a PacketCable network that conforms to PacketCable Event Message Specification version I03.

The *PacketCable Control Point Discovery Interface Specification* document defines an IP-based protocol that can be used to discover a control point for a given IP address. The control point is the place where Quality of Service (QoS) operations, LI content tapping operations, or other operations may be performed.

CISCO ASR 1000 Series Routers

The Cisco ASR 1000 Series Aggregation Services Routers support two types of LI: regular and broadband (per-subscriber). Broadband wiretaps are executed on access subinterfaces and tunnel interfaces. Regular wiretaps are executed on access subinterfaces, tunnel interfaces, and physical interfaces. Wiretaps are not required, and are not executed, on internal interfaces. The router determines which type of wiretap to execute based on the interface that the target's traffic is using.

LI on the Cisco ASR 1000 series routers can intercept traffic based on a combination of one or more of the following fields:

- Destination IP address and mask (IPv4 or IPv6 address)
- Destination port or destination port range
- Source IP address and mask (IPv4 or IPv6 address)
- Source port or source port range
- Protocol ID
- Type of Service (TOS)
- Virtual routing and forwarding (VRF) name, which is translated to a *vrf-tableid* value within the router.
- Subscriber (user) connection ID

The LI implementation on the Cisco ASR 1000 series routers is provisioned using SNMP3 and supports the following functionality:

- RADIUS session intercepts, which can occur in one of the following ways:
  - Interception through Access-Accept packets allows interception to start at the beginning of a session.
  - Interception through CoA-Request packets enables the router to start or stop interception during a session.
- Interception of communication content. The router duplicates each intercepted packet and then places the copy of the packet within a UDP-header encapsulated packet (with a configured CCCid). The router sends the encapsulated packet to the LI mediation device. Even if multiple lawful intercepts are configured on the same data flow, only one copy of the packet is sent to the mediation device. If necessary, the mediation device can duplicate the packet for each LEA.
- Interception of IPv4, IPv4 multicast, IPv6, and IPv6 multicast flows.

VRF Aware LI

VRF Aware LI is the ability to provision a LI wiretap on IPv4 data in a particular Virtual Private Network (VPN). This feature allows a LEA to lawfully intercept targeted data within that VPN. Only IPv4 data within that VPN is subject to the VRF-based LI tap.

VRF Aware LI is available for the following types of traffic:

- ip2ip
- ip2tag (IP to MPLS)
- tag2ip (MPLS to IP)

To provision a VPN-based IPv4 tap, the LI administrative function (running on the mediation device) uses the CISCO-IP-TAP-MIB to identify the name of the VRF table that the targeted VPN uses. The VRF name is used to select the VPN interfaces on which to enable LI in order to execute the tap.

The router determines which traffic to intercept and which mediation device to send the intercepted packets based on the VRF name (along with the source and destination address, source and destination port, and protocol).

Note When using the Cisco-IP-TAP-MIB, if the VRF name is not specified in the stream entry, the global IP routing table is used by default.

Lawful Intercept MIBs

Due to its sensitive nature, the Cisco LI MIBs are only available in software images that support the LI feature. These MIBs are not accessible through the Network Management Software MIBs Support page (http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml).

Restricting Access to the Lawful Intercept MIBs

Only the mediation device and users who need to know about lawful intercepts should be allowed to access the LI MIBs. To restrict access to these MIBs, you must:

1. Create a view that includes the Cisco LI MIBs.
2. Create an SNMP user group that has read-and-write access to the view. Only users assigned to this user group can access information in the MIBs.
3. Add users to the Cisco LI user groups to define who can access the MIBs and any information related to lawful intercepts. Be sure to add the mediation device as a user in this group; otherwise, the router cannot perform lawful intercepts.

For more information, see the Creating a Restricted SNMP View of Lawful Intercept MIBs module.

Note Access to the Cisco LI MIB view should be restricted to the mediation device and to system administrators who need to be aware of lawful intercepts on the router. To access the MIB, users must have level-15 access rights on the router.

RADIUS-Based Lawful Intercept

A RADIUS-based lawful intercept solution enables intercept requests to be sent (through Access-Accept packets or Change of Authorizationt (CoA)-Request packets) to the network access server (NAS) or to the Layer 2 Tunnel Protocol access concentrator (LAC) from the RADIUS server. All traffic data going to or from a PPP or L2TP session is passed to a mediation device. Another advantage of RADIUS-based lawful intercept is the synchronicity of the solution—the tap is set with Access-Accept packets so that all target traffic is intercepted.

Intercept requests are initiated by the mediation device via SNMPv3 messages, and all traffic data going to or from a given IP address is passed to a mediation device. Interception based on IP addresses prevents a session from being tapped until an IP address has been assigned to the session.

The RADIUS-based lawful intercept feature provides High Availability (HA) support for LI for the following modes:

- Access-Accept based LI for the new session
- CoA based LI for existing session

The RADIUS-based LI HA supports only the RADIUS based provisioning. The SNMP-based provisioning is not supported.

Intercept Operation

How Intercept Requests Work Within Access-Accept Packets

When an intercept target begins to establish a connection, an Access-Request packet is sent to the RADIUS server. The RADIUS server responds with an Access-Accept packet containing the four RADIUS attributes.

The NAS or the LAC receives the LI-Action attribute with the value 1, allowing the NAS or LAC to duplicate the traffic data at the start of the new session and forward the duplicated data to the mediation device that was specified through the attributes, MD-IP-Address and MD-Port-Number.

Note If the NAS or LAC cannot start intercepting traffic data for a new session, the session does not get established.

If accounting is enabled (through the **aaa accounting network** command and the **aaa accounting send stop-record authentication failure** command), an Accounting-Stop packet must be sent with the Acct-Termination-Cause attribute (49) set to 15, which means that service is not available.

How Intercept Requests Work Within CoA-Request Packets

After a session has been established for the intercept target, CoA-Request packets can be used for the following tasks:

- Starting the interception of an existing session. The LI-Action attribute is set to 1.
- Stopping the interception of an existing session. The LI-Action attribute is set to 0.
- Issuing a dummy intercept request. The LI-Action attribute is set to 2. The NAS or LAC should not perform any session interception; instead, it searches the session on the basis of the Acct-Session-ID attribute value that was specified in the CoA-Request packets. If a session is found, the NAS or LAC sends a CoA acknowledgment (ACK) response to the RADIUS server. If a session is not found, the NAS or LAC issues a “session not found” error message.

In each case, the RADIUS server must send CoA-Request packets with the identified attributes and the Acct-Session-ID attribute. Each of these attributes must be in the packet.

The Acct-Session-ID attribute identifies the session that will be intercepted. The Acct-Session-ID attribute can be obtained from either the Access-Request packet or the Accounting-Stop packet.

When a session is being tapped and the session terminates, the tap stops. The session does not start when the subscriber logs back in unless the Access-Accept indicates a start tap or a CoA-Request is sent to start the session.

Note The frequency of CoA-Request packets should not exceed a rate of one request every 10 minutes.

Service Independent Intercept (SII)

Cisco developed the Service Independent Intercept (SII) architecture in response to requirements that support lawful intercept for service provider customers. The SII architecture offers well-defined, open interfaces between the Cisco equipment acting as the content Intercept Access Point (IAP) and the mediation device. The modular nature of the SII architecture allows the service provider to choose the most appropriate mediation device to meet specific network requirements and regional, standards-based requirements for the interface to the law enforcement collection function.

The mediation device uses SNMPv3 to instruct the call connect (CC) IAP to replicate the CC and send the content to the mediation device. The CC IAP can be either an edge router or a trunking gateway for voice, and either an edge router or an access server for data.

To increase the security and to mitigate any SNMPv3 vulnerability, the following tasks are required:

Restricting Access to Trusted Hosts (without Encryption)

SNMPv3 provides support for both security models and security levels. A security model is an authentication strategy that is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level will determine the security mechanism employed when handling an SNMP packet.

Additionally, the SNMP Support for the Named Access Lists feature adds support for standard named access control lists (ACLs) to several SNMP commands.

To configure a new SNMP group or a table that maps SNMP users to SNMP views, use the **snmp-server group** command in global configuration mode.

```
access-list my-list permit ip host 10.10.10.1
snmp-server group my-group v3 auth access my-list
```

In this example, the access list named **my-list** allows SNMP traffic only from 10.10.10.1. This access list is then applied to the SNMP group called **my-group**.

Encrypting Lawful Intercept Traffic and Restricting Access to Trusted Hosts

Encryption of intercepted traffic between the router (the content Intercept Access Point (IAP)) and the Mediation Device (MD) is highly recommended.

The following configuration is required:

- Configuring encryption in the router and either an encryption client in the MD or a router associated with the MD to decrypt the traffic.
- Restricting access to trusted hosts.
- Configuring the VPN client.

Configuring encryption in the Router

First configure Authentication, Authorization and Accounting (AAA) parameters. The following example shows how to configure the parameters:

```
aaa authentication login userauthen local
username <username> password 0 <password>
```

The following example uses the internal database; however, external authentication servers can also be specified to perform the authentication.

After configuring the AAA parameters, configure the Internet Security Association and Key Management Protocol (ISAKMP) policy and the crypto map. The following example uses pre-shared keys, Diffie-Hellman (DH) group 2 and AES 256 as the encryption protocol for phase 1 (Internet Key Exchange (IKE)). The crypto map is called dynamic-map and the VPN group is called LI-group. Access-list 108 defines the traffic that is allowed to the router (in this case the ip pool is 10.1.1.1 through 10.1.1.254).

```
crypto isakmp policy 1
encr aes 256
authentication pre-share
group 2
!
crypto isakmp client configuration group LI-group
key <password>
dns 10.10.10.10
wins 10.10.10.20
domain cisco.com
pool ippool
acl 108
!
!
crypto ipsec transform-set myset esp-3des esp-sha-hmac
!
crypto dynamic-map dynmap 10
set transform-set myset
!
!
crypto map clientmap client authentication list userauthen
crypto map clientmap isakmp authorization list groupauthor
crypto map clientmap client configuration address respond
crypto map clientmap 10 ipsec-isakmp dynamic dynmap
!
!
interface GigabitEthernet0/3
ip address <IP address of LI-enabled router> 255.255.255.0
crypto map clientmap
!
!
ip local pool ippool 10.1.1.1 10.1.1.254
!
!
access-list 108 permit ip  10.1.1.0 0.0.0.255 host 10.0.24.4 <IP address of LI-enabled
router>
```

Restricting Access to Trusted Hosts (with Encryption)

The following example shows how to create an ACL that allows only the IP pool (10.1.1.0/24) for VPN clients, and assign that ACL to the SNMPv3 group.

```
access-list my-list permit ip 10.1.1.0 0.0.0.255
snmp-server group my-group v3 auth access my-list
```

Configuring the VPN Client

```
See the
 Installing the VPN Client
 document to download and configure the Cisco VPN Client for Solaris. See the
```

```
Cisco VPN Client installation instructions
 document to download and configure the Cisco VPN Client for other operating systems.
```

How to Configure Lawful Intercept

Although there are no direct user commands to provision lawful intercept on the router, you do need to perform some configuration tasks, such as providing access to LI MIBs, setting up SNMP notifications, and enabling the LI RADIUS session feature. This section describes how to perform the required tasks.

Creating a Restricted SNMP View of Lawful Intercept MIBs

To create and assign users to an SNMP view that includes the Cisco lawful intercept MIBs, perform the steps in this section.

Before you begin

- You must issue the commands in global configuration mode with level-15 access rights.
- SNMPv3 must be configured on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa intercept**
4. **snmp-server view** *view-name MIB-name* **included**
5. **snmp-server view** *view-name MIB-name* **included**
6. **snmp-server view** *view-name MIB-name* **included**
7. **snmp-server group** *group-name* **v3 noauth read** *view-name* **write** *view-name*
8. **snmp-server user** *user-name group-name* **v3 auth md5** *auth-password*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa intercept** **Example:**	Enables lawful intercept on the device.

	Command or Action	Purpose
	`Device(config)# aaa intercept`	• Associate this command with a high administrative security to ensure that unauthorized users cannot stop intercepts if this command is removed. **Note** The **aaa intercept** command is required to set up the wiretap using an IP session.
Step 4	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView ciscoTap2MIB included`	Creates an SNMP view that includes the CISCO-TAP2-MIB (where *exampleView*is the name of the view to create for the MIB). • This MIB is required for both regular and broadband lawful intercept.
Step 5	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView ciscoIpTapMIB included`	Adds the CISCO-IP-TAP-MIB to the SNMP view.
Step 6	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView cisco802TapMIB included`	Adds the CISCO-802-TAP-MIB to the SNMP view.
Step 7	**snmp-server group** *group-name* **v3 noauth read** *view-name* **write** *view-name* **Example:** `Device(config)# snmp-server group exampleGroup v3 noauth read exampleView write exampleView`	Creates an SNMP user group that has access to the LI MIB view and defines the group's access rights to the view.
Step 8	**snmp-server user** *user-name group-name* **v3 auth md5** *auth-password* **Example:** `Device(config)# snmp-server user exampleUser exampleGroup v3 auth md5 examplePassword`	Adds users to the specified user group.
Step 9	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Where to Go Next

The mediation device can now access the lawful intercept MIBs and issue SNMP **set** and **get** requests to configure and run lawful intercepts on the router. To configure the router to send SNMP notification to the mediation device, see the Enabling SNMP Notifications for Lawful Intercept.

Enabling SNMP Notifications for Lawful Intercept

SNMP automatically generates notifications for lawful intercept events. To configure the router to send lawful intercept notifications to the mediation device, perform the steps in this section.

Before you begin

- You must issue the commands in global configuration mode with level-15 access rights.
- SNMPv3 must be configured on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *ip-address* **community-string udp-port** *port notification-type*
4. **snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart** *and* **snmp-server enable traps rf**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *ip-address* **community-string udp-port** *port notification-type* **Example:** `Device(config)# snmp-server 10.2.2.1 community-string udp-port 161 udp`	Specifies the IP address of the mediation device and the password-like community-string that is sent with a notification request. • For lawful intercept, the **udp-port** must be 161 and not 162 (the SNMP default).
Step 4	**snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart** *and* **snmp-server enable traps rf**	Configures the router to send RFC 1157 notifications to the mediation device.

	Command or Action	Purpose

	Example:
	`Device(config)# snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart`
	`Device(config)# snmp-server enable traps rf`
		These notifications indicate authentication failures, link status (up or down), and router restarts.

Step 5	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Disabling SNMP Notifications

To disable SNMP notifications on the router, perform the steps in this section.

Note To disable lawful intercept notifications, use SNMPv3 to set the CISCO-TAP2-MIB object cTap2MediationNotificationEnable to false(2). To reenable lawful intercept notifications through SNMPv3, reset the object to true(1).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no snmp-server enable traps**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no snmp-server enable traps** **Example:** `Device(config)# no snmp-server enable traps`	Disables all SNMP notification types that are available on your system.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling RADIUS Session Intercepts

There are no user CLI commands available to provision the mediation device or taps. However, to enable the intercepts through the CISCO-TAP-MIB you must configure the system to make the account-session-id value available to the mediation device. To enable RADIUS session intercepts on the router, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa intercept**
4. **aaa authentication ppp default group radius**
5. **aaa accounting delay-start all**
6. **aaa accounting send stop-record authentication failure**
7. **aaa accounting network default start-stop group radius**
8. **radius-server attribute 44 include-in-access-req**
9. **radius-server host** *host-name*
10. **aaa server radius dynamic-author**
11. **client** *ip-address*
12. **domain** {**delimiter** *character*| **stripping [right-to-left]**}
13. **server-key** *word*
14. **port** *port-number*
15. **exit**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa intercept**	Enables lawful intercept on the router.

	Command or Action	Purpose
	Example: `Device(config)# aaa intercept`	• Associate this command with a high administrative security to ensure that unauthorized users cannot stop intercepts if this command is removed.
Step 4	**aaa authentication ppp default group radius** **Example:** `Device(config)# aaa authentication ppp default group radius`	Specifies the authentication method to use on the serial interfaces that are running Point-to-Point protocol (PPP). **Note** This command is required because tap information resides only on the RADIUS server. You can authenticate with locally configured information, but you cannot specify a tap with locally configured information.
Step 5	**aaa accounting delay-start all** **Example:** `Device(config)# aaa accounting delay-start all`	Delays the generation of accounting start records until the user IP address is established. Specifying the **all** keyword ensures that the delay applies to all VRF and non-VRF users. **Note** This command is required so that the mediation device can see the IP address assigned to the target.
Step 6	**aaa accounting send stop-record authentication failure** **Example:** `Device(config)# aaa accounting send stop-record authentication failure`	(Optional) Generates accounting stop records for users who fail to authenticate while logging into or during session negotiation. **Note** If a lawful intercept action of 1 does not start the tap, the stop record contains Acct-Termination-Cause, attribute 49, set to 15 (Service Unavailable).
Step 7	**aaa accounting network default start-stop group radius** **Example:** `Device(config)# aaa accounting network default start-stop group radius`	(Optional) Enables accounting for all network-related service requests. **Note** This command is required only to determine the reason why a tap did not start.
Step 8	**radius-server attribute 44 include-in-access-req** **Example:** `Device(config)# radius-server attribute 44 include-in-access-req`	(Optional) Sends RADIUS attribute 44 (Accounting Session ID) in access request packets before user authentication (including requests for preauthentication). **Note** Enter this command to obtain attribute 44 from the Access-Request packet. Otherwise you will have to wait for the accounting packets to be received before you can determine the value of attribute 44.
Step 9	**radius-server host** *host-name* **Example:**	(Optional) Specifies the RADIUS server host.

	Command or Action	Purpose
	`Device(config)# radius-server host host1`	
Step 10	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Configures a device as an Authentication, Authorization, and Accounting (AAA) server to facilitate interaction with an external policy server and enters dynamic authorization local server configuration mode. **Note** This is an optional command if taps are always started with a session starts. The command is required if CoA-Requests are used to start and stop taps in existing sessions.
Step 11	**client** *ip-address* **Example:** `Device(config-locsvr-da-radius)# client 10.0.0.2`	(Optional) Specifies a RADIUS client from which the device will accept CoA-Request packets.

Step 12	**domain** {**delimiter** *character*| **stripping [right-to-left]**}
	Example:
	`Device(config-locsvr-da-radius)# domain stripping right-to-left`
	Example:
	`Device(config-locsvr-da-radius)# domain delimiter @`
		(Optional) Configures username domain options for the RADIUS application.
		• The **delimiter** keyword specifies the domain delimiter. One of the following options can be specified for the *character* argument: **@**, /, **$**, **%**, \, **#** or **-**
		• The **stripping** keyword compares the incoming username with the names oriented to the left of the **@** domain delimiter.
		• The **right-to-left** keyword terminates the string at the first delimiter going from right to left.

Step 13	**server-key** *word* **Example:** `Device(config-locsvr-da-radius)# server-key samplekey`	(Optional) Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 14	**port** *port-number* **Example:** `Device(config-locsvr-da-radius)# port 1600`	(Optional) Specifies a RADIUS client from which the device will accept CoA-Request packets.
Step 15	**exit** **Example:** `Device(config-locsvr-da-radius)# exit`	Exits dynamic authorization local server configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuring Circuit ID Based Tapping

To configure circuit ID based tapping of user session data packets and RADIUS authentication data packets on the router, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **subscriber access pppoe unique-key circuit-id**
4. **end**
5. **show pppoe session all**
6. **show idmgr session key circuit-id** *circuit-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**subscriber access pppoe unique-key circuit-id** **Example:** `Device(config)#subscriber access pppoe unique-key circuit-id`	Specifies a unique circuit ID tag for a PPPoE user session to be tapped on the router.
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 5	**show pppoe session all** **Example:**	Displays the circuit-id tag in the PPPoE session, which is used in the next step to verify the user session.

	Command or Action	Purpose
	`Device# show pppoe session all`	

Step 6	**show idmgr session key circuit-id** *circuit-id*
	Example:
	`Device# show idmgr session key circuit-id Ethernet4/0.100:PPPoE-Tag-1`
	Example:
	Example:
	`session-handle = AA000007`
	Example:
	`aaa-unique-id = 0000000E`
	Example:
	`circuit-id-tag = Ethernet4/0.100:PPPoE-Tag-1`
	Example:
	`interface = nas-port:0.0.0.0:0/1/1/100`
	Example:
	`authen-status = authen`
	Example:
	`username = user1@cisco.com`
	Example:
	`addr = 106.1.1.3`
	Example:
	`session-guid = 650101020000000E`
	Example:
	`The session hdl AA000007 in the record is valid`
	Example:
	`The session hdl AA000007 in the record is valid`
	Example:
	`No service record found`
		Verifies the user session information in the ID Manager (IDMGR) database by specifying the unique circuit ID tag.

Configuration Examples for Lawful Intercept

Example: Enabling Mediation Device Access Lawful Intercept MIBs

The following example shows how to enable the mediation device to access the lawful intercept MIBs. It creates an SNMP view (tapV) that includes four LI MIBs (CISCO-TAP2-MIB, CISCO-IP-TAP-MIB, CISCO-802-TAP-MIB, and CISCO-USER-CONNECTION-TAP-MIB). It also creates a user group that has read, write, and notify access to MIBs in the tapV view.

```
aaa intercept
snmp-server view tapV ciscoTap2MIB included
snmp-server view tapV ciscoIpTapMIB included
snmp-server view tapV cisco802TapMIB included
snmp-server view tapV ciscoUserConnectionTapMIB included
snmp-server group tapGrp v3 noauth read tapV write tapV notify tapV
snmp-server user MDuser tapGrp v3 auth md5 MDpasswd
snmp-server engineID local 1234
```

Example: Enabling RADIUS Session Lawful Intercept

The following example shows the configuration of a RADIUS-Based Lawful Intercept solution on a router acting as a network access server (NAS) device employing an Ethernet PPP connection over Ethernet (PPPoE) link:

```
aaa new-model
!
aaa intercept
!
aaa group server radius SG
server 10.0.56.17 auth-port 1645 acct-port 1646
!
aaa authentication login LOGIN group SG
aaa authentication ppp default group SG
aaa authorization network default group SG
aaa accounting send stop-record authentication failure
aaa accounting network default start-stop group SG
!
aaa server radius dynamic-author
client 10.0.56.17 server-key cisco
!
vpdn enable
!
bba-group pppoe PPPoE-TERMINATE
virtual-template 1
!
interface Loopback0
ip address 10.1.1.2 255.255.255.0
!
interface GigabitEthernet4/1/0
description To RADIUS server
ip address 10.0.56.20 255.255.255.0
duplex auto
!
interface GigabitEthernet4/1/2
```

```
description To network
ip address 10.1.1.1 255.255.255.0
duplex auto
!
interface GigabitEthernet5/0/0
description To subscriber
no ip address
!
interface GigabitEthernet5/0/0.10
encapsulation dot1q 10
protocol pppoe group PPPoE-TERMINATE
!
interface Virtual-Template1
ip unnumbered Loopback0
ppp authentication chap
!
radius-server attribute 44 include-in-access-req
radius-server attribute nas-port format d
radius-server host 10.0.56.17 auth-port 1645 acct-port 1646
radius-server key cisco
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring SNMP Support	*Configuring SNMP Support*
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
PacketCable™ Control Point Discovery Interface Specification	*PacketCable ™ Control Point Discovery Interface Specification* (PKT-SP-CPD-I02-061013)

MIBs

MIB	MIBs Link
• CISCO-TAP2-MIB • CISCO-IP-TAP-MIB • CISCO-802-TAP-MIB • CISCO-USER-CONNECTION-TAP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC-2865	*Remote Authentication Dial In User Service (RADIUS)*
RFC-3576	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*
RFC-3924	*Cisco Architecture for Lawful Intercept in IP Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Lawful Intercept

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 237: Feature Information for Lawful Intercept

Feature Name	Releases	Feature Information
Lawful Intercept	Cisco IOS XE Release 2.4 Cisco IOS XE Release 3.15S	The Lawful Intercept (LI) feature supports service providers in meeting the requirements of law enforcement agencies to provide the ability to intercept VoIP or data traffic going through the edge routers. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.15S, the Lawful Intercept feature was introduced on tunnel interfaces for the Cisco ASR 1000 Series Aggregation Services Routers.
VRF Aware LI (Lawful Intercept)	Cisco IOS XE Release 2.4	VRF Aware LI is the ability to provision a LI wiretap on IPv4 data in a particular Virtual Private Network (VPN). In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Feature Name	Releases	Feature Information
RADIUS-based Lawful Intercept	Cisco IOS XE Release 2.4 Cisco IOS XE Release 3.5S	The LI implementation is provisioned using SNMP3 and supports RADIUS session intercepts. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.5, High Availability support was added for RADIUS-Based Lawful Intercept.
Circuit ID based tapping of PPP session for Lawful Intercept.	Cisco IOS XE Release 2.5	In Cisco IOS XE Release 2.5, circuit ID based tapping of a PPP session is introduced. Circuit ID based tapping works only if the tap is provisioned after the user session is active. It is assumed in this instance that the user session is uniquely identified by a circuit ID tag.
Circuit ID based tapping for Lawful Intercept	Cisco IOS XE Release 2.6	In Cisco IOS XE Release 2.6, pre-provisioning of circuit-ID based tapping of a PPP session is introduced. If the tap is provisioned before a user session is active, then the tap is effective whenever the user session becomes active. Also, corresponding RADIUS authentication and accounting packets are tapped. It is assumed in this instance that the user session is uniquely identified by a circuit ID tag.
Non-Lawful Intercept (Non-LI) Images	Cisco IOS XE Release 3.10S	In Cisco IOS XE Release 3.10S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The Non-LI images will be available from Cisco IOS XE Release 3.10S onwards and will not contain the LI subsystems.

CHAPTER 175

LI Support for IPoE Sessions

The LI Support for IPoE Sessions feature extends support for provisioning lawful intercept (LI) to IP over Ethernet (IPoE) sessions in accordance with RFC 2866. This document describes RADIUS-based LI for IPoE. See the "Lawful Intercept Architecture" module for information on LI architecture and components and for configuration tasks and examples.

Restrictions for LI Support for IPoE Sessions

The following restrictions apply to RADIUS-based LI for IPoE sessions:

- You cannot use Access-Accept packets to start TAP for a RADIUS proxy session when the LI parameters are encrypted.
- The **aaa intercept** command must be configured to accept attribute value pairs (AVPs) associated with RADIUS-based LI. The frequency of change of authentication (CoA) requests to start, stop, or no-action, should not exceed a rate of 1 per 10 minutes.
- Intercepted traffic from different users is sent to the same mediation device (MD). You must use a unique stream ID (made up of the first four digits of the eight-digit intercept ID) for each MD.
- The format of intercepted packets captured using RADIUS-based LI include the L2 header; this is different from the format of SNMP-based LI.
- Per-flow tapping is not supported through RADIUS-based LI; it is supported with SNMP-based LI.

Additional References for LI Support for IPoE Sessions

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Configuring SNMP support	*Configuring SNMP Support*
Security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
PacketCable™ Control Point Discovery Interface Specification	*PacketCable ™ Control Point Discovery Interface Specification* (PKT-SP-CPD-I02-061013)

MIBs

MIB	MIBs Link
• CISCO-IP-TAP-MIB • CISCO-TAP2-MIB • CISCO-802-TAP-MIB • CISCO-USER-CONNECTION-TAP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2866	*RADIUS Accounting*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for LI Support for IPoE Sessions

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 238: Feature Information for LI Support for IPoE Sessions

Feature Name	Releases	Feature Information
LI Support for IPoE Sessions	Cisco IOS XE Release 3.10S	Extends support for provisioning LI to IPoE sessions in accordance with RFC 2866.

CHAPTER 176

Image Verification

The Image Verification feature allows users to automatically verify the integrity of Cisco IOS XE images and provisioning files. Thus, users can be sure that an image or provisioning file is protected from accidental corruption, which can occur at any time during transit, starting from the moment the files are generated by Cisco until they reach the user.

Restrictions for Image Verification

Image Verification is applied to and attempted on any file; however, if the file is not an image file or provisioning file, image verification will not occur and you will see the following error, "SIGNATURE-4-NOT_PRESENT."

Note The Image Verification feature can only be used to check the integrity of a Cisco IOS XE software image or provisioning file that is stored on a Cisco IOS XE device. It cannot be used to check the integrity of an image on a remote file system or an image running in memory.

Information About Image Verification

Note Throughout this document, any references to Cisco IOS XE images, also applies to provisioning files.

Benefits of Image Verification

The efficiency of Cisco IOS XE routers is improved because the routers can now automatically detect when the integrity of an image or provisioning file is accidentally corrupted as a result of transmission errors or disk corruption.

How Image Verification Works

Because a production image undergoes a sequence of transfers before it is copied into the memory of a router, the integrity of the image is at risk of accidental corruption every time a transfer occurs. When downloading an image from Cisco.com, a user can run a message-digest5 (MD5) hash on the downloaded image and verify that the MD5 digest posted on Cisco.com is the same as the MD5 digest that is computed on the user's server. However, many users choose not to run an MD5 digest because it is 128-bits long and the verification is manual. Image verification allows the user to automatically validate the integrity of all downloaded images, thereby, significantly reducing user interaction.

How to Use Image Verification

Globally Verifying the Integrity of an Image

The **file verify auto** command enables image verification globally; that is, all images that are to be copied (via the **copy** command) or reloaded (via the **reload** command) are automatically verified. Although both the **copy** and **reload** commands have a **/verify** keyword that enables image verification, you must issue the keyword each time you want to copy or reload an image. The **file verify auto** command enables image verification by default, so you no longer have to specify image verification multiple times.

If you have enabled image verification by default but prefer to disable verification for a specific image copy or reload, the **/noverify** keyword, along with either the **copy** or the **reload** command, will override the **file verify auto** command.

Use this task to enable automatic image verification.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **file verify auto**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**file verify auto** **Example:** `Device(config)# file verify auto`	Enables automatic image verification.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode. You must exit global configuration mode if you are going to copy or reload an image.

What to Do Next

After issuing the **file verify auto** command, you do not have to issue the **/verify** keyword with the **copy** or the **reload** command because each image that is copied or reloaded will be automatically verified.

Verifying the Integrity of an Image That Is About to Be Copied

When issuing the **copy** command, you can verify the integrity of the copied file by entering the **/verify** keyword. If the integrity check fails, the copied file will be deleted. If the file that is about to be copied does not have an embedded hash (an old image), you will be prompted whether or not to continue with the copying process. If you choose to continue, the file will be successfully copied; if you choose not to continue, the copied file will be deleted.

Without the **/verify** keyword, the **copy** command could copy a file that is not valid. Thus, after the **copy** command has been successfully executed, you can issue the **verify** command at any time to check the integrity of the files that are in the storage of the router.

Use this task to verify the integrity of an image before it is copied onto a router.

SUMMARY STEPS

1. **enable**
2. **copy** [**/erase**] [**/verify**| **/noverify**] *source-url destination-url*
3. **verify** [**/md5** [*md5-value*]] *filesystem: file-url*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**copy** [**/erase**] [**/verify**| **/noverify**] *source-url destination-url* **Example:** `Device# copy /verify tftp://10.1.1.1/cat3k_caa-universalk9.SSA.16.1.0.EFT3-1.bin flash:`	Copies any file from a source to a destination. • **/verify** --Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify** --Does not verify the signature of the destination file before the image is copied. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.
Step 3	**verify** [**/md5** [*md5-value*]] *filesystem: file-url*] **Example:** `Device# flash: tftp://10.1.1.1/cat3k_caa-universalk9.SSA.16.1.0.EFT3-1.bin flash:`	(Optional) Verifies the integrity of the images in the Device's storage.

Verifying the Integrity of an Image That Is About to Be Reloaded

By issuing the **reload** command with the /**verify** keyword, the image that is about to be loaded onto your system will be checked for integrity. If the /**verify** keyword is specified, image verification will occur before the system initiates the reboot. Thus, if verification fails, the image will not be loaded.

Note Because different platforms obtain the file that is to be loaded in various ways, the file specified in BOOTVAR will be verified. If a file is not specified, the first file on each subsystem will be verified. On certain platforms, because of variables such as the configuration register, the file that is verified may not be the file that is loaded.

Use this task to verify the integrity of an image before it is reloaded onto a router.

SUMMARY STEPS

1. **enable**
2. **reload** [[**warm**] [**/verify**| **/noverify**] *text* | [**warm**] [**/verify**| **/noverify**] **in** [*hh* **:** *mm* [*text*] | [**warm**] [**/verify**| **/noverify**] **at** *hh* **:** *mm* [*month day* | *day month*] [*text*] | [**warm**] [**/verify**| **/noverify**] **cancel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**reload** [[**warm**] [**/verify**| **/noverify**] *text*	[**warm**] [**/verify**| **/noverify**] **in** [*hh* **:** *mm* [*text*]	[**warm**] [**/verify**| **/noverify**] **at** *hh* **:** *mm* [*month day*	*day month*] [*text*]	[**warm**] [**/verify**| **/noverify**] **cancel**] **Example:** `Device# reload /verify`	Reloads the operating system. • /**verify**--Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify** --Does not verify the signature of the destination file before the image is reloaded. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.

Configuration Examples for Image Verification

Global Image Verification Example

The following example shows how to enable automatic image verification. After enabling this command, image verification will automatically occur for all images that are either copied (via the **copy** command) or reloaded (via the **reload** command).

```
Device(config)# file verify auto
```

Image Verification via the copy Command Example

The following example shows how to specify image verification before copying an image:

```
Device# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:
Destination filename [c7200-js-mz]?
Accessing tftp://10.1.1.1/jdoe/c7200-js-mz...
Loading jdoe/c7200-js-mz from 10.1.1.1 (via FastEthernet0/0):!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 19879944 bytes]
19879944 bytes copied in 108.632 secs (183003 bytes/sec)
Verifying file integrity of disk0:/c7200-js-mz
.....................................................................................
.....................................................................................
.....................................................................................
.........................Done!
Embedded Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash       MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash               MD5 :44A7B9BDDD9638128C35528466318183
Signature Verified
```

Image Verification via the reload Command Example

The following example shows how to specify image verification before reloading an image onto the Device:

```
Device# reload /verify
Verifying file integrity of bootflash:c7200-kboot-mz.121-8a.E
%ERROR:Signature not found in file bootflash:c7200-kboot-mz.121-8a.E.
Signature not present. Proceed with verify? [confirm]
Verifying file disk0:c7200-js-mz
.....................................................................
.....................................................Done!
Embedded Hash     MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash      MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash          MD5 :44A7B9BDDD9638128C35528466318183
Signature Verified
Proceed with reload? [confirm]n
```

Verify Command Sample Output Example

The following example shows how to specify image verification via the **verify** command:

```
Device# verify disk0:c7200-js-mz
%Filesystem does not support verify operations
Verifying file integrity of disk0:c7200-js-mz.....................................
.....................................................................Done!
Embedded Hash     MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash      MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash          MD5 :44A7B9BDDD9638128C35528466318183
```

Signature Verified

Additional References

The following sections provide references related to the Image Verification feature.

Related Documents

Related Topic	Document Title
Configuration tasks and information for loading, maintaining, and rebooting system images	Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide
Additional commands for loading, maintaining, and rebooting system images	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Image Verification

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 239: Feature Information for Image Verification

Feature Name	Releases	Feature Information
Image Verification		The Image Verification feature allows users to automatically verify the integrity of Cisco IOS XE images. The following commands were introduced or modified: **copy**, **file verify auto**, **reload**, **verify**.

PART XVII

IPsec Data Plane

CHAPTER 177

IPsec Anti-Replay Window Expanding and Disabling

Cisco IP security (IPsec) authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. The decryptor keeps track of which packets it has seen on the basis of these numbers. Currently, the default window size is 64 packets. Generally, this number (window size) is sufficient, but there are times when you may want to expand this window size. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

Prerequisites for IPsec Anti-Replay Window Expanding and Disabling

- Before configuring this feature, you should have already created a crypto map or crypto profile.
- To configure the IPsec Anti-Replay Window: Expanding and Disabling feature, you should understand the following concept: IPsec Anti-Replay Window, on page 2318

Information About IPsec Anti-Replay Window Expanding and Disabling

IPsec Anti-Replay Window

Cisco IPsec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association [SA] anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value X of the highest sequence number that it has already seen. N is the window size, and the decryptor also remembers whether it has seen packets having sequence numbers from X-N+1 through X. Any packet with the sequence number X-N is discarded. Currently, N is set at 64, so only 64 packets can be tracked by the decryptor.

At times, however, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded when they arrive outside of the 64 packet replay window at the decryptor. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

Increasing the anti-replay window size has no impact on throughput and security. The impact on memory is insignificant because only an extra 128 bytes per incoming IPsec SA is needed to store the sequence number on the decryptor. It is recommended that you use the full 1024 window size to eliminate any future anti-replay problems.

How to Configure IPsec Anti-Replay Window Expanding and Disabling

Configuring IPsec Anti-Replay Window Expanding and Disabling Globally

To configure IPsec Anti-Replay Window: Expanding and Disabling globally (so that it affects all SAs that are created), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association replay window-size** [*N*]
4. **crypto ipsec security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association replay window-size** [*N*] **Example:** `Router (config)# crypto ipsec security-association replay window-size 256`	Sets the size of the SA replay window globally. **Note** Configure this command or the **crypto ipsec security-association replay disable** command. The two commands are not used at the same time.
Step 4	**crypto ipsec security-association replay disable** **Example:** `Router (config)# crypto ipsec security-association replay disable`	Disables checking globally. **Note** Configure this command or the **crypto ipsec security-association replay window-size** command. The two commands are not used at the same time.

Configuring IPsec Anti-Replay Window Expanding and Disabling on a Crypto Map

To configure IPsec Anti-Replay Window: Expanding and Disabling on a crypto map so that it affects those SAs that have been created using a specific crypto map or profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**]
4. **set security-association replay window-size** [*N*]
5. **set security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router (config)# crypto map ETHO 17 ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps.
Step 4	**set security-association replay window-size** [*N*] **Example:** `Router (crypto-map)# set security-association replay window-size 128`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or cyrpto profile. **Note** Configure this command or the **set security-association replay disable** command. The two commands are not used at the same time.
Step 5	**set security-association replay disable** **Example:** `Router (crypto-map)# set security-association replay disable`	Disables replay checking for a particular crypto map, dynamic crypto map, or crypto profile. **Note** Configure this command or the **set security-association replay window-size** command. The two commands are not used at the same time.

Troubleshooting Tips

- If your replay window size has not been set to a number that is high enough for the number of packets received, you will receive a system message such as the following:

```
*Nov 17 19:27:32.279: %CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed connection
id=1
```

The above message is generated when a received packet is judged to be outside the anti-replay window.

Configuration Examples for IPsec Anti-ReplayWindow Expanding and Disabling

Global Expanding and Disabling of an Anti-Replay Window Example

The following example shows that the anti-replay window size has been set globally to 1024:

```
version 2.1
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!
boot-start-marker
boot-end-marker
!
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
!
ip audit po max-events 100
no ftp-server write-enable
!
!
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key cisco123 address 192.165.201.2 !
crypto ipsec security-association replay window-size 1024 !
crypto ipsec transform-set basic esp-des esp-md5-hmac !
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252  serial restart-delay 0  crypto map mymap !
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255 access-list 101 remark
 Crypto ACL
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
```

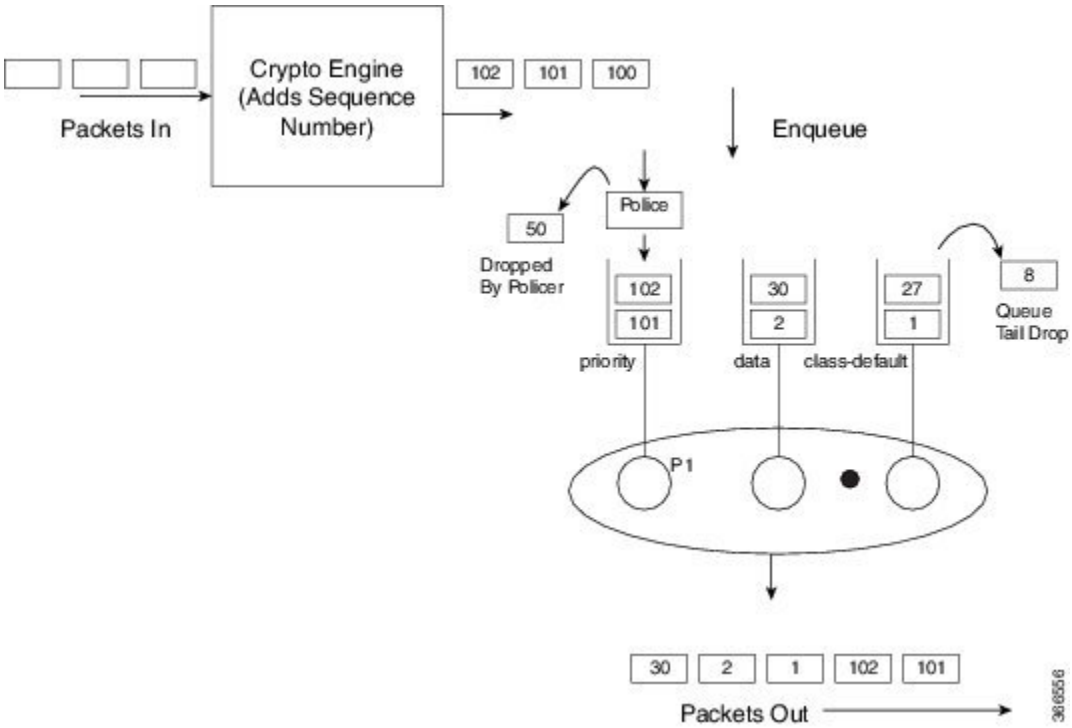
Expanding and Disabling of an Anti-Replay Window for Crypto Maps or Crypto Profiles Example

The following example shows that anti-replay checking is disabled for IPsec connections to 172.17.150.2 but enabled (and the default window size is 64) for IPsec connections to 172.17.150.3 and 172.17.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname networkserver1
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1 enable password ww !
ip subnet-zero
!
cns event-service server
crypto isakmp policy 1
authentication pre-share
crypto isakmp key cisco170 address 172.17.150.2 crypto isakmp key cisco180 address
172.17.150.3 crypto isakmp key cisco190 address 172.17.150.4
crypto ipsec transform-set 170cisco esp-des esp-md5-hmac crypto ipsec transform-set 180cisco
 esp-des esp-md5-hmac crypto ipsec transform-set 190cisco esp-des esp-md5-hmac
crypto map ETH0 17 ipsec-isakmp
 set peer 172.17.150.2
 set security-association replay disable set transform-set 170cisco match address 170 crypto
 map ETH0 18 ipsec-isakmp  set peer 192.168.1.3 set transform-set 180cisco match address
180 crypto map ETH0 19 ipsec-isakmp set peer 192.168.1.4 set transform-set 190cisco match
address 190 !
interface FastEthernet0
 ip address 172.17.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.16.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.18.170.0 255.255.255.0 172.17.150.2 ip route 172.19.180.0 255.255.255.0
172.17.150.3 ip route 172.20.190.0 255.255.255.0 172.17.150.4 no ip http server !
access-list 170 permit ip 172.16.160.0 0.0.0.255 172.18.170.0 0.0.0.255 access-list 180
permit ip 172.16.160.0 0.0.0.255 172.19.180.0 0.0.0.255 access-list 190 permit ip 172.16.160.0
 0.0.0.255 172.20.190.0 0.0.0.255 !
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
logi
end
```

IPsec Anti Replay Mechanism for QoS

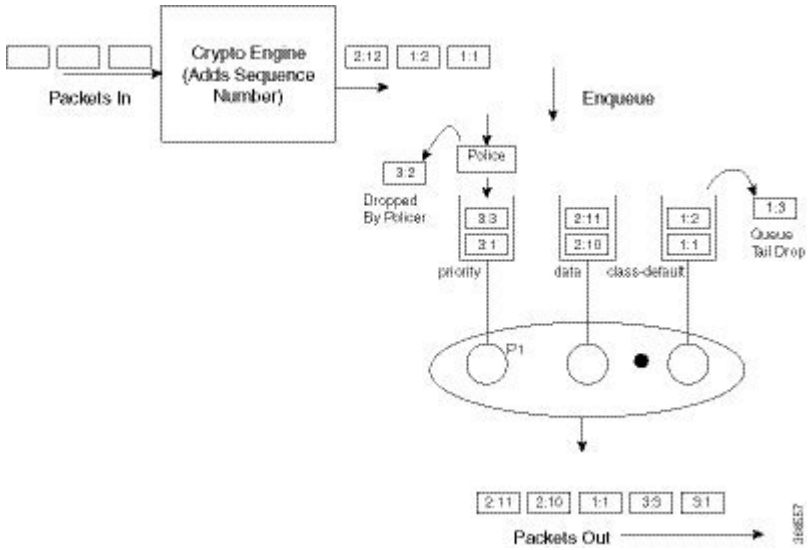
It is normal for packets to be reordered in IP networks, where QoS mechanisms (on the egress interface of the encrypting device or on other network elements in the path), loadbalancing mechanisms or routing / path selection mechanisms (that send different flows over different paths) are used.



The above diagram shows how anti-replay protection system causes problems when QoS reorders packets. The encryption engine adds sequence numbers. After these numbers are added, packets are enqueued in egress queues depending on the application within that packet. In the example in the diagram, packets are already present in the bandwidth queues (data and class-default), when packets with the sequence numbers 101 and 102 are enqueued in the priority queue. The priority packets will be scheduled first. When the decrypting device receives the packet with the sequence number 101, the history in the sliding window is moved to 101, implying that the sliding window creates a history of sequence numbers 30-101. When the next packet which has the sequence number 102 is received, the history in the sliding window is changed to 39-102. Now, that there are no more packets in the priority queue, packets from one of the other queue is taken – for example, packet with the sequence number 1. Although this is the first time the decrypting device is receiving a packet with sequence number 1, the packet is dropped because of the history maintained in the sliding window.

Moving QoS scheduling before the encryption may solve the anti-replay issue but would render the QoS functionality useless. In addition, scheduling needs to be driven by the congestion of the egress interface (or a shaper on that interface). Increasing the size of the anti-replay window places a huge load load on the memory of the devices that handles this functionality.

Hence, the solution of maintaining multiple sequence number spaces per security association was introduced. The number spaces would be aligned with the egress queuing scheme such that all packets in a given queue would receive a sequence number from the same sequence number space. Since all packets within a sequence number space would go through the same queue, the possibility of egress QoS causing reordering within those packets is eliminated. It is still possible (but unlikely) that reordering within a number space could happen elsewhere in the network. If packets are tail dropped rather than enqueued out of sequence (not out of order), sequence numbers will still be received on the receiving side. Hence, we still maintain a history window per sequence number space but that history is considerably shorter.



The image shows that the sequence number consists of two parts, namely the selector and the sequence number. The receiving side would use the selector to choose the correct history to use and the sequence number would operate as always.

Note IPSec Anti-Replay feature does not support Group Encrypted Transport VPN (GETVPN) when multiple sequence number space (multi-SNS) is enabled.

IPsec Anti-Replay Packet Loss Avoidance

The IPsec Anti-Replay Packet Loss Avoidance feature avoids unnecessary IPSec Anti-Replay packet drops when QoS is configured with IPSec. However, some packet drops can happen under certain circumstances when QoS is used together with IPSec Anti-Replay enabled. Anti-Replay drops are seen for a second or two with multi-SNS enabled when a class-map is added or removed while crypto interface is attached on the peer router. The traffic recovers after a couple of seconds and no drops are seen after that.

The Anti-Replay drops can occur in the following situations:

- When a packet is in transit, a class is deleted from the QoS policymap. The packets that belong to this class are exhausted and the incoming packets are queued behind all the packets in the class-default queue. This can cause disruption in the sequence number space causing Anti-Replay drops. The queue becomes empty and the system recovers soon enough to resume normal behavior.
- When an ESP-based High Availability is configured and the over-subscribed traffic is sent through all the sequence number spaces Anti-Replay drops occur. With over-subscribed traffic on the sender side, traffic is shaped based on QoS policy. As a result, the receiving router gets packets with out of order sequence numbers. These drops are momentary and are recovered soon.
- During rekeying of security associations (SA), a router keeps both the old and new inbound Security Parameter Index (SPI) for a short period of time. Old SA is deleted after a short period. After the old SA is deleted, if router receives any packet with old SPI (which can happen when there is a QoS policy), it drops the packet with invalid SPI error.

Configuring IPsec Anti-Replay for QoS

Given below is the command to enable multiple sequence number space per IPSec SA:

```
Device(config)#crypto ipsec security-association multi-sn
```

Caution All existing sessions need to be cleared before configuring this feature. Else, traffic from the existing sessions will be dropped.

Caution This feature needs to be configured on both the tunnel routers in an IPSec connection. If this featues is only enabled on one router, the other router will drop packets.

Show Commands

show platform hardware qfp active feature ipsec datapath crypto-sa

This command displays the mapping between the sequence number spaces and the sequence numbers in an IPsec SA in QFP:

```
Device# show platform hardware qfp active feature ipsec datapath crypto-sa 4
Crypto Context Handle: e8b06b60
peer sa handle: 0
anti-replay enabled
esn disabled
Outbound SA
Total SNS: 16
Space                current seq number
----------------------------------------
   0                                   0
   1                                   0
   2                                   0
   3                                   0
   4                                   0
   5                                   0
   6                                   0
   7                                   0
   8                                   0
   9                                   0
  10                                   0
  11                                 100
  12                                   0
  13                                   0
  14                                   0
  15                                   0
```

show platform hardware qfp active feature ipsec sa

This command displays the IPsec SA in Cisco QuantumFlow Processor (Cisco QFP):

```
Device# show platform hardware qfp active feature ipsec sa 6
QFP ipsec sa Information

         QFP sa  id: 6
```

```
          pal sa id: 170
         QFP spd id: 1
          QFP sp id: 2
            QFP spi: 0xa4a5244(172642884)
        crypto ctx: 0x00000000e8b14a20
              flags: 0x4640068 (Details below)
                   : src:IKE  valid:Yes  soft-life-expired:No  hard-life-expired:No
                   : replay-check:No  proto:ESP  mode:Receive-only  direction:Egress
                  : qos_preclassify:No  qos_group:No
                   : frag_type:AFTER_ENCRYPT  df_bit_type:COPY
                   : sar_enable:No  getvpn_mode:SNDRCV_SA
                   : doing_translation:No  assigned_outside_rport:No
                   : inline_tagging_enabled:No
          qos_group: 0x0
                mtu: 0x59e=1438
            mtu_adj: 0x588=1416
          sar_delta: 0
         sar_window: 0x0
         sibling_sa: 0x0
             sp_ptr: 0xe8abc000
            sbs_ptr: 0xe8a73878
     local endpoint: 33.0.0.3
    remote endpoint: 33.0.0.4
   cgid.cid.fid.rid: 1.1.1.11141121
              ivrf: 0
              fvrf: 0
    trans udp sport: 0
    trans udp dport: 0
    first intf name: Tunnel0
nat fixup src port: 0
    nat fixup  ip: 0.0.0.0
```

show platform software ipsec fp active flow

This command displays the IPsec SA in the fman-fp process for a given flow ID:

```
Device# show platform software ipsec fp active flow identifier 169
Flow id: 169
                mode: tunnel
           direction: inbound
            protocol: esp
                 SPI: 0xbcd8840
      local IP addr: 33.0.0.3
     remote IP addr: 33.0.0.4
   crypto device id: 0
      crypto map id: 1
             SPD id: 1
         QFP SPD id: 1
    ACE line number: 1
      QFP SA handle: 5
IOS XE interface id: 11
     interface name: Tunnel0
   Crypto SA ctx id: 0x00000000e8b148c0
             cipher: AES-128
               auth: SHA256
initial seq.number: 0
      timeout, mins: 0
              flags: exp time;exp traffic;
Time limits

    soft limit(sec): 3401
    hard limit(sec): 3568
Traffic limits
```

```
     soft limit(kb): 3962880
     hard limit(kb): 4608000
     inline_tagging: DISABLED
anti-replay window: 64
SPI Selector:

   remote addr low: 0.0.0.0
  remote addr high: 0.0.0.0
    local addr low: 33.0.0.3
   local addr high: 33.0.0.3
Classifier: range

   src IP addr low: 33.0.0.3
  src IP addr high: 33.0.0.3
   dst IP addr low: 33.0.0.4
  dst IP addr high: 33.0.0.4
      src port low: 0
     src port high: 65535
      dst port low: 0
     dst port high: 65535
      protocol low: 47
     protocol high: 47
------------- Statistics

            octets(delta): 0
      total octets(delta): 4718576880
           packets(delta): 0
   dropped packets(delta): 0
      replay drops(delta): 0
      auth packets(delta): 0
        auth fails(delta): 0
encrypted packets(delta): 0
     encrypt fails(delta): 0
---------- End statistics

       object state: active
--------------- AOM

         cpp aom id: 894
         cgm aom id: 0
          n2 aom id: 891
          if aom id: 0
```

show crypto ipsec sa <ip> peer

This command retrieves the IPSec SA ID for the given peer and displays the SA in all the layers, which is from the IOS layer to the QFP layer.

```
Device# polaris-csr#show crypto ipsec sa peer 33.0.0.4 platform

interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 33.0.0.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (33.0.0.3/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (33.0.0.4/255.255.255.255/47/0)
   current_peer 33.0.0.4 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 190, #pkts encrypt: 190, #pkts digest: 190
    #pkts decaps: 190, #pkts decrypt: 190, #pkts verify: 190
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
```

```
    #send errors 0, #recv errors 0

     local crypto endpt.: 33.0.0.3, remote crypto endpt.: 33.0.0.4
     plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet2
     current outbound spi: 0xA4A5244(172642884)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0xBCD8840(198019136)
        transform: esp-aes esp-sha256-hmac ,
        in use settings ={Tunnel, }
        conn id: 2169, flow_id: CSR:169, sibling_flags FFFFFFFF80004048, crypto map:
Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4607985/3255)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xA4A5244(172642884)
        transform: esp-aes esp-sha256-hmac ,
        in use settings ={Tunnel, }
        conn id: 2170, flow_id: CSR:170, sibling_flags FFFFFFFF80004048, crypto map:
Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4607989/3255)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:

     outbound pcp sas:
```

Additional References

The following sections provide references related to IPsec Anti-Replay Window: Expanding and Disabling.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Security Command Reference
IP security and encryption	Configuring Security for VPNs with IPsec

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IPsec Anti-Replay Window Expanding and Disabling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 240: Feature Information for IPsec Anti-Replay Window: Expanding and Disabling

Feature Name	Releases	Feature Information
IPsec Anti-Replay Window: Expanding and Disabling	Cisco IOS XE Release 2.1 IPsec Anti-Replay Window Expanding and Disabling, on page 2317	The following commands were introduced or modified: **crypto ipsec security-association replay disable**, **ipsec security-association replay window-size**, **security-association replay disable**, **security-association replay window-size**.
IPSec anti-replay should work when QoS is enabled in CSR platforms.	Cisco IOS XE Release 16.6.1	This feature enables support for IPSec anti-replay mechanism when QoS is enabled in Cisco Cloud Services Router 1000V Series. The following commands were introduced or modified: **show platform hardware qfp active feature ipsec**, **show platform software ipsec fp active flow**, **show crypto ipsec sa**.
IPSec anti-replay should work when QoS is enabled in ISR 4300/4200 platforms.	Cisco IOS XE Release 16.7.1	This feature ensures that IPSec anti-replay mechanism works when QoS is enabled in ISR platforms except ISR 44xx.
Anti-replay QoS/IPSec packet loss avoidance	Cisco IOS XE Release 16.8.1	This feature avoids IPSec anti-replay packet drops when QoS is used with IPSec anti-replay enabled. This support is added on Octeon-based ASR platforms only.

CHAPTER 178

Pre-Fragmentation for IPsec VPNs

The Pre-Fragmentation for IPsec VPNs feature increases performance between Cisco IOS XE routers and VPN clients by delivering encryption throughput at maximum encryption hardware accelerator speeds for packets that are near the maximum transmission unit (MTU) size. Packets are fragmented into equally sized units to prevent further downstream fragmentation.

Restrictions for Pre-Fragmentation for IPsec VPNs

Take the following information into consideration before this feature is configured:

- Pre-fragmentation for IPsec VPNs operates in IPsec tunnel mode and IPsec tunnel mode with GRE, but not with IPsec transport mode.
- Pre-fragmentation for IPsec VPNs configured on the decrypting router in a unidirectional traffic scenario does not improve the performance or change the behavior of either of the peers.
- Pre-fragmentation for IPsec VPNs occurs before the transform is applied if compression is turned on for outgoing packets.
- Pre-fragmentation for IPsec VPNs functionality depends on the egress interface **crypto ipsec df-bit** configuration and the incoming packet "do not fragment" (DF) bit state. See the table below.

Table 241: Pre-Fragmentation for IPsec VPNs Dependencies

Pre-Fragmentation for IPsec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Enabled	crypto ipsec df-bit clear	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit clear	1	Fragmentation occurs before encryption.

Pre-Fragmentation for IPsec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Disabled	crypto ipsec df-bit clear	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit clear	1	Fragmentation occurs after encryption and packets are reassembled before decryption.
Enabled	crypto ipsec df-bit set	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit set	1	Packets are dropped.
Disabled	crypto ipsec df-bit set	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit set	1	Packets are dropped.
Enabled	crypto ipsec df-bit copy	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit copy	1	Packets are dropped.
Disabled	crypto ipsec df-bit copy	0	Fragmentation occurs after encryption, and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit copy	1	Packets are dropped.

Information About Pre-Fragmentation for IPsec VPNs

Pre-fragmentation for IPsec VPNs

When a packet is nearly the size of the MTU of the outbound link of the encrypting router and it is encapsulated with IPsec headers, it is likely to exceed the MTU of the outbound link. This causes packet fragmentation after encryption. The decrypting router must then reassemble these packets in the process path, which decreases the decrypting router's performance.

The Pre-fragmentation for IPsec VPNs feature increases the decrypting router's performance by enabling it to operate in the high-performance CEF path instead of the process path. An encrypting router can predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPsec security association (SA). If it is predetermined that the packet exceeds the MTU of the output interface, the packet is fragmented before encryption. This function avoids process-level reassembly before decryption and helps improve decryption performance and overall IPsec traffic throughput.

Note The pre-fragmentation feature is turned off by default for tunnel interfaces. To receive pre-fragmentation performance benefits, turn pre-fragmentation on after ensuring that the tunnel interfaces have the same MTU on both ends.

Crypto maps are no longer used to define fragmentation behavior that occurred before and after encryption. Now, IPsec Virtual Tunnel Interface (also referred to as Virtual-Template interface) (VTI) fragmentation behavior is determined by the IP MTU settings that are configured on the VTI.

See the IPsec Virtual Tunnel Interface feature document for more information on VTIs.

Note If fragmentation after-encryption behavior is desired, then set the VTI IP MTU to a value that is greater than the egress router interface IP MTU. Use the **show ip interface tunnel**command to display the IP MTU value.

How to Configure Pre-Fragmentation for IPsec VPNs

Configuring Pre-Fragmentation for IPsec VPNs

Perform this task to configure Pre-Fragmentation for IPsec VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip mtu** *bytes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config-if)# interface tunnel0`	Specifies the interface on which the VTI is configured and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1500` **Example:**	Specifies the VTI MTU size in bytes of IP packets on the egress interface for IPsec VPNs. **Note** If after-encryption fragmentation behavior is desired, then set the VTI IP MTU to a value that is greater than the egress router interface IP MTU. Use the **show ip interface tunnel**command to display the IP MTU value.

Additional References

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference
IPsec	IPsec Virtual Tunnel Interface feature document

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Pre-Fragmentation for IPsec VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 242: Feature Information for Pre-Fragmentation for IPsec VPNs

Feature Name	Releases	Feature Information
Pre-Fragmentation for IPsec VPNs	Cisco IOS XE 2.1	This feature increases performance between Cisco IOS routers and VPN clients by delivering encryption throughput at maximum encryption hardware accelerator speeds for packets that are near the maximum transmission unit (MTU) size. Packets are fragmented into equally sized units to prevent further downstream fragmentation. The following command was introduced or modified: **ip mtu (interface configuration) .**

CHAPTER 179

Invalid Security Parameter Index Recovery

When an invalid security parameter index error (shown as “Invalid SPI”) occurs in IP Security (IPsec) packet processing, the Invalid Security Parameter Index Recovery feature allows for an Internet Key Exchange (IKE) security association (SA) to be established. The “IKE” module sends notification of the “Invalid SPI” error to the originating IPsec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed.

Prerequisites for Invalid Security Parameter Index Recovery

Before configuring the Invalid Security Parameter Index Recovery feature, you must have enabled IKE and IPsec on your router.

Restrictions for Invalid Security Parameter Index Recovery

If an IKE SA is being initiated to notify an IPsec peer of an “Invalid SPI” error, there is the risk that a denial-of-service (DoS) attack can occur. The Invalid Security Parameter Index Recovery feature has a built-in mechanism to minimize such a risk, but because there is a risk, the Invalid Security Parameter Index Recovery feature is not enabled by default. You must enable the command using command-line interface (CLI).

Information About Invalid Security Parameter Index Recovery

How the Feature Works

An IPsec “black hole” occurs when one IPsec peer “dies” (for example, a peer can “die” if a reboot occurs or if an IPsec peer somehow gets reset). Because one of the peers (the receiving peer) is completely reset, it loses

its IKE SA with the other peer. Generally, when an IPsec peer receives a packet for which it cannot find an SA, it tries to send an IKE "INVALID SPI NOTIFY" message to the data originator. This notification is sent using the IKE SA. If there is no IKE SA available, the receiving peer drops the packet.

Note A single SA has only two peers. However, a SADB can have multiple SAs, whereby each SA has an association with a different peer.

When an invalid security parameter index (SPI) is encountered, the Invalid Security Parameter Index feature provides for the setting up of an IKE SA with the originator of the data, and the IKE "INVALID SPI NOTIFY" message is sent. The peer that originated the data "sees" the "INVALID SPI NOTIFY" message and deletes the IPsec SA that has the invalid SPI. If there is further traffic from the originating peer, there will not be any IPsec SAs, and new SAs will be set up. Traffic will flow again. The default behavior (that is, without configuring the Invalid Security Parameter Index Recovery feature) is that the data packet that caused the invalid SPI error is dropped. The originating peer keeps on sending the data using the IPsec SA that has the invalid SPI, and the receiving peer keeps dropping the traffic (thus creating the "black hole").

The IPsec module uses the IKE module to send an IKE "INVALID SPI NOTIFY" message to the other peer. Once the invalid SPI recovery is in place, there should not be any significant dropping of packets although the IPsec SA setup can itself result in the dropping of a few packets.

To configure your router for the Invalid Security Parameter Index Recovery feature, use the **crypto isakmp invalid-spi-recovery** command. The IKE SA will not be initiated unless you have configured this command.

How to Configure Invalid Security Parameter Index Recovery

Configuring Invalid Security Parameter Index Recovery

To configure the Invalid Security Parameter Index Recovery feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp invalid-spi-recovery**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**crypto isakmp invalid-spi-recovery** **Example:** `Router (config)# crypto isakmp invalid-spi-recovery`	Initiates the IKE module process whereby the IKE module notifies the receiving peer that an "Invalid SPI" error has occurred.

Verifying a Preshared Configuration

To determine the status of the IPsec SA for traffic between two peers, you can use the **show crypto ipsec sa** command. If the IPsec SA is available on one peer and not on the other, there is a "black hole" situation, in which case you will see the invalid SPI errors being logged for the receiving peer. If you turn console logging on or check the syslog server, you will see that these errors are also being logged.

The daigram below shows the topology of a typical preshared configuration setup. Host 1 is the initiating peer (initiator), and Host 2 is the receiving peer (responder).

Figure 89: Preshared Configuration Topology

SUMMARY STEPS

1. Initiate the IKE and IPsec SAs between Host 1 and Host 2
2. Clear the IKE and IPsec SAs on Router B
3. Send traffic from Host 1 to Host 2 and ensure that new IKE and IPsec SAs are correctly established
4. Check for an invalid SPI message on Router B

DETAILED STEPS

Step 1 Initiate the IKE and IPsec SAs between Host 1 and Host 2

Router A

Example:

```
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       / 10.2.2.2              10.1.1.1        QM_IDLE               1       0
```

Router B

Example:

```
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.1.1.1       10.2.2.2        QM_IDLE               1       0
```

Router A

Example:

```
Router# show crypto ipsec sa interface fastethernet0/0
interface: FastEthernet0/0
    Crypto map tag: testtag1, local addr. 10.1.1.1
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   current_peer: 10.2.2.2:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 10.1.1.1, remote crypto endpt.: 10.2.2.2
     path mtu 1500, media mtu 1500
     current outbound spi: 7AA69CB7
     inbound esp sas:
      spi: 0x249C5062(614223970)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
      spi: 0xB16D1587(2976716167)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        replay detection support: Y
     inbound pcp sas:
     outbound esp sas:
      spi: 0x7AA69CB7(2057739447)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3595)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
      spi: 0x1214F0D(18960141)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3594)
        replay detection support: Y
     outbound pcp sas:
```

Router B

Example:

```
Router# show crypto ipsec sa interface Fastethernet1/0
interface: FastEthernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
```

```
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 249C5062
     inbound esp sas:
      spi: 0x7AA69CB7(2057739447)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4421281/3593)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
      spi: 0x1214F0D(18960141)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4421281/3593)
        replay detection support: Y
     inbound pcp sas:
     outbound esp sas:
      spi: 0x249C5062(614223970)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4421285/3593)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
      spi: 0xB16D1587(2976716167)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4421285/3592)
        replay detection support: Y
     outbound pcp sas:
```

Step 2 Clear the IKE and IPsec SAs on Router B

Example:

```
Router# clear crypto isakmp
Router# clear crypto sa
Router# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.2.2.       10.1.1.1      MM_NO_STATE           1       0 (deleted)
Router# show crypto ipsec sa
interface: FastEthernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
```

```
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 0
     inbound esp sas:
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
     outbound ah sas:
     outbound pcp sas:
```

Step 3 Send traffic from Host 1 to Host 2 and ensure that new IKE and IPsec SAs are correctly established

Example:

```
ping
Protocol [ip]: ip
Target IP address: 10.0.2.2
Repeat count [5]: 30
Datagram size [100]: 100
Timeout in seconds [2]:
Extended commands [n]: no
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 30, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
..!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 93 percent (28/30), round-trip min/avg/max = 1/3/8 ms
RouterB# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.1.1.1       10.2.2.2     QM_IDLE              3       0
       /        10.1.1.1       10.2.2.2      MM_NO_STATE          1       0 (deleted)
RouterB# show crypto ipsec sa
interface: FastEthernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
    #pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: D763771F
     inbound esp sas:
      spi: 0xE7AB4256(3886760534)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5127, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        IV size: 8 bytes
        replay detection support: Y
```

```
     inbound ah sas:
      spi: 0xF9205CED(4179647725)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5125, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        replay detection support: Y
     inbound pcp sas:
     outbound esp sas:
      spi: 0xD763771F(3613619999)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5128, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3596)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
      spi: 0xEB95406F(3952427119)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5126, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3595)
        replay detection support: Y
     outbound pcp sas:
RouterA# show crypto isakmp sa
  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.2.2       10.1.1.1      MM_NO_STATE           1       0 (deleted)
       /        10.2.2.2       10.1.1.1      QM_IDLE               2       0
```

Step 4 Check for an invalid SPI message on Router B

Example:

```
Router# show logging
Syslog logging: enabled (10 messages dropped, 13 messages rate-limited, 0 flushes, 0 overruns, xml
disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: level debugging, 43 messages logged, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged
Log Buffer (8000 bytes):
*Mar 24 20:55:45.739: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
        destaddr=10.2.2.2, prot=51, spi=0x1214F0D(18960141), srcaddr=10.1.1.1
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #2,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:47.743: IPSEC(key_engine): got a queue event with 2 kei messages
```

```
*Mar 24 20:55:47.743: IPSEC(spi_response): getting spi 4179647725 for SA
        from 10.2.2.2        to 10.1.1.1     for prot 2
*Mar 24 20:55:47.747: IPSEC(spi_response): getting spi 3886760534 for SA
        from 10.2.2.2     to 10.1.1.1        for prot 3
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524099
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524100
*Mar 24 20:55:48.135: IPSEC(key_engine): got a queue event with 4 kei messages
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xF9205CED(4179647725), conn_id= 939529221, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xEB95406F(3952427119), conn_id= 939529222, keysize= 0, flags= 0xA
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xE7AB4256(3886760534), conn_id= 939529223, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xD763771F(3613619999), conn_id= 939529224, keysize= 0, flags= 0xA
*Mar 24 20:55:48.139: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:48.139: IPSEC(mtree_add_ident): src 10.2.2.2, dest 10.1.1.1, dest_port 0
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 51,
    sa_spi= 0xF9205CED(4179647725),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529221
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 51,
    sa_spi= 0xEB95406F(3952427119),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529222
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0xE7AB4256(3886760534),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529223
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 50,
    sa_spi= 0xD763771F(3613619999),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529224
ipseca-72a#
```

Configuration Examples for Invalid SecurityParameter Index Recovery

Invalid Security Parameter Index Recovery Example

The following example shows that invalid security parameter index recovery has been configured on Router A and Router B. The following example shows the topology used for this example.

Router A

```
Router# show running-config
Building configuration...
Current configuration : 2048 bytes
!
version 2.1
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service tcp-small-servers
!
hostname ipseca-71a
!
logging queue-limit 100
no logging console
enable secret 5 $1$4GZB$L2YOmnenOCNAu0jgFxebT/
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.2.2.2
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
```

```
 set peer 10.2.2.2
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.0.0.0
 no ip route-cache cef
 duplex full
 speed 100
 crypto map testtag1
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.0.0.0
 no ip route-cache cef
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.3.3.3 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.0.1 host 10.0.2.2
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
```

```
call rsvp-sync
!
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
 login
!
!
end
ipseca-71a#
```

Router B

```
Router# show running-config
Building configuration...
Current configuration : 2849 bytes
!
version 2.1
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname ipseca-72a
!
logging queue-limit 100
no logging console
enable secret 5 $1$kKqL$5Th5Qhw1ubDkkK90KWFxi1
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.1.1.1
```

```
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 10.2.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
 crypto map testtag1
!
interface FastEthernet1/1
 ip address 10.0.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
!
interface FastEthernet1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface FastEthernet1/6
 no ip address
 no ip route-cache
 no ip mroute-cache
```

```
 shutdown
 duplex half
!
interface FastEthernet1/7
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial3/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial3/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.2.2 host 10.0.0.1
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
```

```
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
!
end
```

Additional References

The following sections provide references relate to Invalid Security Parameter Index Recovery.

Related Documents

Related Topic	Document Title
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Invalid Security ParameterIndex Recovery

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 243: Feature Information for Invalid Security Parameter Index Recovery

Feature Name	Releases	Feature Information
Invalid Special Parameter Index (SPI) Recovery	Cisco IOS XE Release 2.1	When an invalid SPI occurs in IPsec packet processing, the Invalid Security Parameter Index Recovery feature allows for an IKE SA to be established. The “IKE” module sends notification of the “Invalid SPI” error to the originating IPsec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed. The following command was introduced or modified: **crypto isakmp invalid-spi-recovery.**

CHAPTER 180

IPsec Dead Peer Detection Periodic Message Option

The IPsec Dead Peer Detection Periodic Message Option feature allows you to configure your router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers.

Prerequisites for IPsec Dead Peer Detection Periodic Message Option

Before configuring the IPsec Dead Peer Detection Periodic Message Option feature, you should have the following:

- Familiarity with configuring IP Security (IPsec).
- An IKE peer that supports DPD (dead peer detection). Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS XE software in all modes of operation--site-to-site and Easy VPN server.

Restrictions for IPsec Dead Peer Detection Periodic Message Option

Using periodic DPD potentially allows the router to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.

Information About IPsec Dead Peer Detection Periodic Message Option

How DPD and Cisco IOS XE Keepalive Features Work

DPD and Cisco IOS XE keepalives function on the basis of the timer. If the timer is set for 10 seconds, the router will send a "hello" message every 10 seconds (unless, of course, the router receives a "hello" message from the peer). The benefit of IOS keepalives and periodic DPD is earlier detection of dead peers. However, IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD also has an on-demand approach. The contrasting on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a router has to send outbound traffic and the liveliness of the peer is questionable, the router sends a DPD message to query the status of the peer. If a router has no traffic to send, it never sends a DPD message. If a peer is dead, and the router never has any traffic to send to the peer, the router will not find out until the IKE or IPsec security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the router is not trying to communicate with the peer). On the other hand, if the router has traffic to send to the peer, and the peer does not respond, the router will initiate a DPD message to determine the state of the peer.

Using the IPsec Dead Peer Detection Periodic Message Option

With the IPsec Dead Peer Detection Periodic Message Option feature, you can configure your router so that DPD messages are "forced" at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a router has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the router does not have to wait until the IKE SA times out to find out.

If you want to configure the DPD periodic message option, you should use the **crypto isakmp keepalive** command with the **periodic** keyword. If you do not configure the **periodic** keyword, the router defaults to the on-demand approach.

Note When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.

Using DPD and Cisco IOS XE Keepalive Featureswith Multiple Peers in the Crypto Map

DPD and Cisco IOS XE keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the router to detect a dead IKE peer, and when the router detects the dead state, the router deletes the IPsec and IKE SAs to the peer. If you configure multiple peers, the router will switch over to the next listed peer for a stateless failover.

How to Configure IPsec Dead Peer Detection Periodic Message Option

Configuring a Periodic DPD Message

To configure a periodic DPD message, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**] **Example:** `Router (config)# crypto isakmp keepalive 10 periodic`	Allows the gateway to send DPD messages to the peer. • *seconds* --When the **periodic** keyword is used, this argument is the number of seconds between DPD messages; the range is from 10 to 3600 seconds. When the **on-demand** keyword is used, this argument is the number of seconds during which traffic is not received from the peer before DPD retry messages are sent if there is data (IPSec) traffic to send; the range is from 10 to 3600 seconds. **Note** If you do not specify a time interval, an error message appears. • *retry-seconds* --(Optional) Number of seconds between DPD retry messages if the DPD retry message is missed by the peer; the range is from 2 to 60 seconds. Once 1 DPD message is missed by the peer, the router moves to a more aggressive state and sends the DPD retry

	Command or Action	Purpose
		message at the faster retry interval, which is the number of seconds between DPD retries if the DPD message is missed by the peer. The default DPD retry message is sent every 2 seconds. Five aggressive DPD retry messages can be missed before the tunnel is marked as down. **Note** To configure DPD with IPsec High Availability (HA), the recommendation is to use a value other than the default (which is 2 seconds). A keepalive timer of 10 seconds with 5 retries seems to work well with HA because of the time that it takes for the router to get into active mode. • **periodic** --(Optional) DPD messages are sent at regular intervals. • **on-demand** --(Optional) The default behavior. DPD retries are sent on demand. **Note** Because this option is the default, the **on-demand** keyword does not appear in configuration output.

Configuring DPD and Cisco IOS XE Keepalives with Multiple Peersin the Crypto Map

To configure DPD and IOS keepalives to be used in conjunction with the crypto map to allow for stateless failover, perform the following steps. This configuration will cause a router to cycle through the peer list when it detects that the first peer is dead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set peer** {*host-name* [**dynamic**] | *ip-address*}
5. **set transform-set** *transform-set-name*
6. **match address** [*access-list-id* | *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto map green 1 ipsec-isakmp`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • The **ipsec-isakmp** keyword indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 4	**set peer** {*host-name* [**dynamic**] | *ip-address*} **Example:** `Router (config-crypto-map)# set peer 10.12.12.12`	Specifies an IPsec peer in a crypto map entry. • You can specify multiple peers by repeating this command.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set txfm`	Specifies which transform sets can be used with the crypto map entry. • You can specify more than one transform set name by repeating this command.
Step 6	**match address** [*access-list-id* | *name*] **Example:** `Router (config-crypto-map)# match address 101`	Specifies an extended access list for a crypto map entry.

Verifying That DPD Is Enabled

DPD allows the router to clear the IKE state when a peer becomes unreachable. If DPD is enabled and the peer is unreachable for some time, you can use the **clear crypto session** command to manually clear IKE and IPsec SAs.

The **debug crypto isakmp** command can be used to verify that DPD is enabled.

SUMMARY STEPS

1. **enable**
2. **clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]] | [**fvrf** *vrf-name*] [**ivrf** *vrf-name*]
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]] | [**fvrf** *vrf-name*] [**ivrf** *vrf-name*] **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPsec and IKE SAs).
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuration Examples for IPsec Dead Peer Detection Periodic Message Option

Site-to-Site Setup with Periodic DPD Enabled Example

The following configurations are for a site-to-site setup with periodic DPD enabled. The configurations are for the IKE Phase 1 policy and for the IKE preshared key.

IKE Phase 1 Policy

```
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
!
```

IKE Preshared Key

```
crypto isakmp key kd94j1ksldz address 10.2.80.209 255.255.255.0
crypto isakmp keepalive 10 periodic
crypto ipsec transform-set Trans1 esp-aes esp-sha-hmac

!
!
interface
  ip address 10.1.32.14 255.255.255.0
  speed auto

!
```

Verifying DPD Configuration Using the debug crypto isakmp Command Example

The following sample output from the **debug crypto isakmp** command verifies that IKE DPD is enabled:

```
*Mar 25 15:17:14.131: ISAKMP:(0:1:HW:2):IKE_DPD is enabled, initializing timers
```

To see that IKE DPD is enabled (and that the peer supports DPD): when periodic DPD is enabled, you should see the following debug messages at the interval specified by the command:

```
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2):purging node 899852982 *Mar 25 15:18:52.111:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:18:52.111: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to sending the DPD R_U_THERE message.

```
*Mar 25 15:18:52.123: ISAKMP (0:268435457): received packet from 10.2.80.209
dport 500 sport 500 Global (I) QM_IDLE
*Mar 25 15:18:52.123: ISAKMP: set new node -443923643 to QM_IDLE *Mar 25 15:18:52.131:
ISAKMP:(0:1:HW:2): processing HASH payload. message ID =
-443923643
*Mar 25 15:18:52.131: ISAKMP:(0:1:HW:2): processing NOTIFY R_U_THERE_ACK protocol 1
 spi 0, message ID = -443923643, sa = 81BA4DD4
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2): DPD/R_U_THERE_ACK received from peer
10.2.80.209, sequence 0x9
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):deleting node -443923643 error FALSE
reason "informational (in) state 1"
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_INFO_NOTIFY *Mar
25 15:18:52.135: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to receiving the acknowledge (ACK) message from the peer.

```
Router#
*Mar 25 15:47:35.335: ISAKMP: set new node -90798077 to QM_IDLE *Mar 25 15:47:35.343:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:35.343: ISAKMP:(0:1:HW:2):purging node -90798077 *Mar 25 15:47:35.347:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:47:35.347: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:36.611: ISAKMP:(0:1:HW:2):purging node 1515050537 *Mar 25 15:47:37.343:
ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:37.343: ISAKMP: set new node -1592471565 to QM_IDLE *Mar 25 15:47:37.351:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:37.351: ISAKMP:(0:1:HW:2):purging node -1592471565 *Mar 25 15:47:37.355:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:37.355: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:39.355: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:39.355: ISAKMP: set new node 1758739401 to QM_IDLE *Mar 25 15:47:39.363:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
```

```
*Mar 25 15:47:39.363: ISAKMP:(0:1:HW:2):purging node 1758739401 *Mar 25 15:47:39.367:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:39.367: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:41.367: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:41.367: ISAKMP: set new node 320258858 to QM_IDLE *Mar 25 15:47:41.375:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):purging node 320258858 *Mar 25 15:47:41.379:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:43.379: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:43.379: ISAKMP: set new node -744493014 to QM_IDLE *Mar 25 15:47:43.387:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:43.387: ISAKMP:(0:1:HW:2):purging node -744493014 *Mar 25 15:47:43.391:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:43.391: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):peer 10.2.80.209 not responding! *Mar 25 15:47:45.391:
 ISAKMP:(0:1:HW:2):peer does not do paranoid keepalives.
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.395: ISAKMP: Unlocking IPSEC struct 0x81E5C4E8 from
delete_siblings, count 0
*Mar 25 15:47:45.395: %CRYPTO-5-SESSION_STATUS: Crypto tunnel is DOWN.  Peer
10.2.80.209:500       Id: 10.2.80.209
*Mar 25 15:47:45.399: ISAKMP: set new node -2061951065 to QM_IDLE *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):purging node -2061951065 *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_DEST_SA
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.415: ISAKMP: Unlocking IKE struct 0x81E5C4E8 for
isadb_mark_sa_deleted(), count 0
*Mar 25 15:47:45.415: ISAKMP: Deleting peer node by peer_reap for 10.2.80.209:
81E5C4E8
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -1067612752 error TRUE
reason "peers alive"
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -114443536 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node 2116015069 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node -1981865558 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):Input = IKE_MESG_INTERNAL, IKE_PHASE1_DEL *Mar 25
15:47:45.419: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
*Mar 25 15:47:45.419: ISAKMP: received ke message (4/1)
*Mar 25 15:47:45.419: ISAKMP: received ke message (3/1)
*Mar 25 15:47:45.423: ISAKMP: ignoring request to send delete notify (no ISAKMP
sa) src 10.1.32.14 dst 10.2.80.209 for SPI 0x3A7B69BF
```

```
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting SA reason "" state (I)
MM_NO_STATE (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -1067612752 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -114443536 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node 2116015069 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):deleting node -1981865558 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH *Mar 25
15:47:45.427: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
```

The above message shows what happens when the remote peer is unreachable. The router sends one DPD R_U_THERE message and four retransmissions before it finally deletes the IPsec and IKE SAs.

DPD and Cisco IOS XE Keepalives Used in Conjunction with Multiple Peers in a Crypto Map Example

The following example shows that DPD and Cisco IOS XE keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE will be used to establish the security associations (SAs). In this example, an SA could be set up to the IPsec peer at 10.0.0.1, 10.0.0.2, or 10.0.0.3.

```
crypto isakmp keepalive 10 periodic
crypto map green 1 ipsec-isakmp
  set peer 10.0.0.1
  set peer 10.0.0.2
  set peer 10.0.0.3
  set transform-set txfm
  match address 101
```

Additional References

The following sections provide references related to IPsec Dead Peer Detection Periodic Message Option.

Related Documents

Related Topic	Document Title
Configuring IPsec	Configuring Security for VPNs with IPsec
IPsec commands	Cisco IOS Security Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
DPD conforms to the Internet draft "draft-ietf-ipsec-dpd-04.txt," which is pending publication as an Informational RFC (a number has not yet been assigned).	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Dead Peer Detection Periodic Message Option

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 244: Feature Information for Dead Peer Detection

Feature Name	Releases	Feature Information
Dead Peer Detection Periodic Message Option	Cisco IOS XE Release 2.1	This feature allows you to configure your router to query the liveliness of its IKE peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers. The following command was introduced or modified: **crypto isakmp keepalive**.

CHAPTER 181

IPsec NAT Transparency

The IPsec NAT Transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPsec.

Restrictions for IPsec NAT Transparency

Although this feature addresses many incompatibilities between NAT and IPsec, the following problems still exist:

Internet Key Exchange (IKE) IP Address and NAT

This incompatibility applies only when IP addresses are used as a search key to find a preshared key. Modification of the IP source or destination addresses by NAT or reverse NAT results in a mismatch between the IP address and the preshared key.

Embedded IP Addresses and NAT

Because the payload is integrity protected, any IP address enclosed within IPsec packets cannot be translated by NAT. Protocols that use embedded IP addresses include FTP, Internet Relay Chat (IRC), Simple Network Management Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), H.323, and Session Initiation Protocol (SIP).

Information About IPsec NAT Transparency

Benefit of IPsec NAT Transparency

Before the introduction of this feature, a standard IPsec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPsec packet. This feature makes NAT IPsec-aware, thereby, allowing remote access users to build IPsec tunnels to home gateways.

Feature Design of IPsec NAT Traversal

The IPsec NAT Transparency feature introduces support for IPsec traffic to travel through NAT or PAT points in the network by encapsulating IPsec packets in a User Datagram Protocol (UDP) wrapper, which allows the packets to travel across NAT devices. The following sections define the details of NAT traversal:

IKE Phase 1 Negotiation NAT Detection

During Internet Key Exchange (IKE) phase 1 negotiation, two types of NAT detection occur before IKE Quick Mode begins--NAT support and NAT existence along the network path.

To detect NAT support, you should exchange the vendor identification (ID) string with the remote peer. During Main Mode (MM) 1 and MM 2 of IKE phase 1, the remote peer sends a vendor ID string payload to its peer to indicate that this version supports NAT traversal. Thereafter, NAT existence along the network path can be determined.

Detecting whether NAT exists along the network path allows you to find any NAT device between two peers and the exact location of NAT. A NAT device can translate the private IP address and port to public value (or from public to private). This translation changes the IP address and port if the packet goes through the device. To detect whether a NAT device exists along the network path, the peers should send a payload with hashes of the IP address and port of both the source and destination address from each end. If both ends calculate the hashes and the hashes match, each peer knows that a NAT device does not exist on the network path between them. If the hashes do not match (that is, someone translated the address or port), then each peer needs to perform NAT traversal to get the IPsec packet through the network.

The hashes are sent as a series of NAT discovery (NAT-D) payloads. Each payload contains one hash; if multiple hashes exist, multiple NAT-D payloads are sent. In most environments, there are only two NAT-D payloads--one for the source address and port and one for the destination address and port. The destination NAT-D payload is sent first, followed by the source NAT-D payload, which implies that the receiver should expect to process the local NAT-D payload first and the remote NAT-D payload second. The NAT-D payloads are included in the third and fourth messages in Main Mode and in the second and third messages in Aggressive Mode (AM).

IKE Phase 2 Negotiation NAT Traversal Decision

While IKE phase 1 detects NAT support and NAT existence along the network path, IKE phase 2 decides whether or not the peers at both ends will use NAT traversal. Quick Mode (QM) security association (SA) payload in QM1 and QM2 is used to for NAT traversal negotiation.

Because the NAT device changes the IP address and port number, incompatibilities between NAT and IPsec can be created. Thus, exchanging the original source address bypasses any incompatibilities.

UDP Encapsulation of IPsec Packets for NAT Traversal

In addition to allowing IPsec packets to traverse across NAT devices, UDP encapsulation also addresses many incompatibility issues between IPsec and NAT and PAT. The resolved issues are as follows:

Incompatibility Between IPsec ESP and PAT Resolved

If PAT found a legislative IP address and port, it would drop the Encapsulating Security Payload (ESP) packet. To prevent this scenario, UDP encapsulation is used to hide the ESP packet behind the UDP header. Thus, PAT treats the ESP packet as a UDP packet, processing the ESP packet as a normal UDP packet.

Incompatibility Between Checksums and NAT Resolved

In the new UDP header, the checksum value is always assigned to zero. This value prevents an intermediate device from validating the checksum against the packet checksum, thereby, resolving the TCP UDP checksum issue because NAT changes the IP source and destination addresses.

Incompatibility Between Fixed IKE Destination Ports and PAT Resolved

PAT changes the port address in the new UDP header for translation and leaves the original payload unchanged.

To see how UDP encapsulation helps to send IPSec packets see the figures below.

Figure 90: Standard IPsec Tunnel Through a NAT/PAT Point (No UDP Encapsulation)

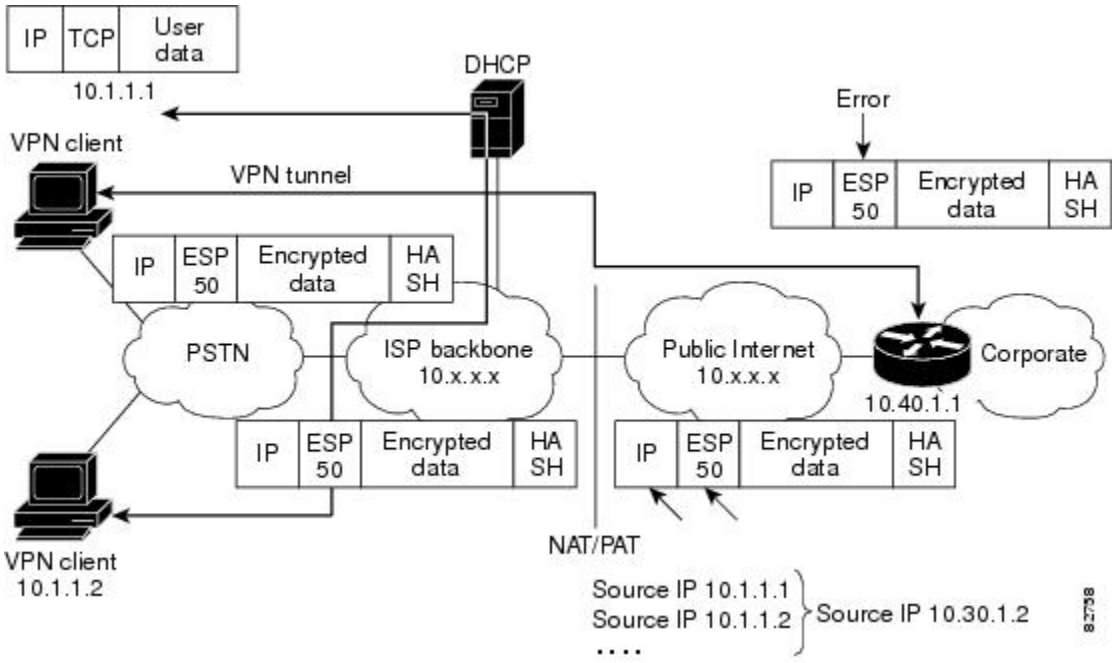
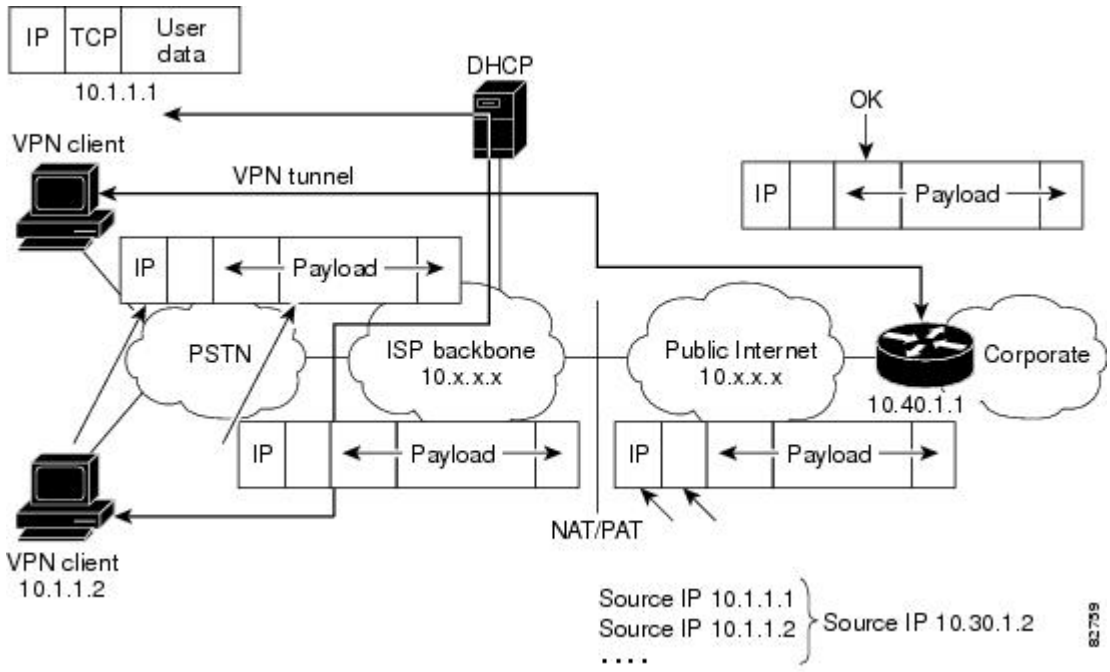


Figure 91: IPsec Packet with UDP Encapsulation



UDP Encapsulated Process for Software Engines Transport Mode and Tunnel Mode ESP Encapsulation

After the IPsec packet is encrypted by a hardware accelerator or a software crypto engine, a UDP header and a non-IKE marker (which is 8 bytes in length) are inserted between the original IP header and ESP header. The total length, protocol, and checksum fields are changed to match this modification. The first figure below shows an IPsec packet before and after transport mode is applied; the second figure below shows an IPsec packet before and after tunnel mode is applied.

Figure 92: Transport Mode--IPsec Packet Before and After ESP Encapsulation

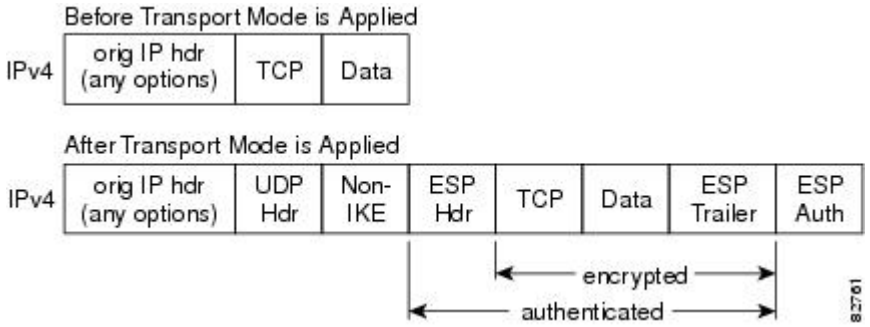
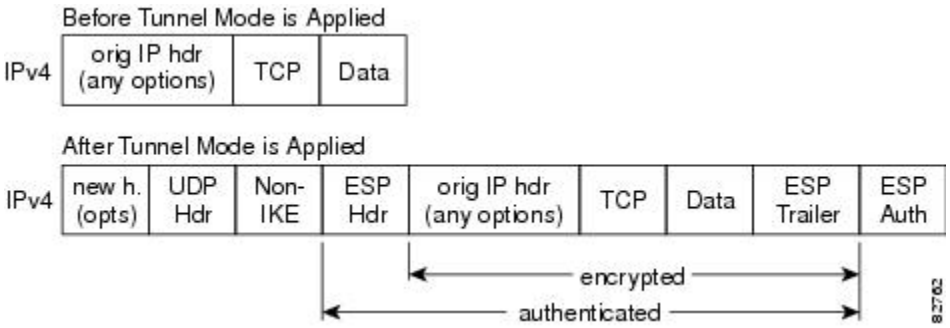


Figure 93: Tunnel Mode--IPsec Packet Before and After ESP Encapsulation



NAT Keepalives

NAT keepalives are enabled to keep the dynamic NAT mapping alive during a connection between two peers. NAT keepalives are UDP packets with an unencrypted payload of 1 byte. Although the current dead peer detection (DPD) implementation is similar to NAT keepalives, there is a slight difference: DPD is used to detect peer status, while NAT keepalives are sent if the IPsec entity did not send or receive the packet at a specified period of time--valid range is between 5 to 3600 seconds.

If NAT keepalives are enabled (via the **crypto isakmp nat keepalive** command), users should ensure that the idle value is shorter than the NAT mapping expiration time, which is 20 seconds.

How to Configure NAT and IPsec

Configuring NAT Traversal

NAT Traversal is a feature that is auto detected by VPN devices. There are no configuration steps for a router running Cisco IOS XE Release 2.1. If both VPN devices are NAT-T capable, NAT Traversal is auto detected and auto negotiated.

Disabling NAT Traversal

You may wish to disable NAT traversal if you already know that your network uses IPsec-awareness NAT (spi-matching scheme). To disable NAT traversal, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto ipsec nat-transparency udp-encapsulation**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**no crypto ipsec nat-transparency udp-encapsulation**
	Example:
	`Router(config)#`
	`no crypto ipsec nat-transparency udp-encapsulation`
		Disables NAT traversal.

Configuring NAT Keepalives

To configure your router to send NAT keepalives, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp nat keepalive** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**crypto isakmp nat keepalive** *seconds*
	Example:
	`Router(config)#`
	`crypto isakmp nat keepalive 20`
		Allows an IPsec node to send NAT keepalive packets.
		• *seconds* --The number of seconds between keepalive packets; range is between 5 to 3,600 seconds.

	Command or Action	Purpose
		Note When the timer is modified, it is modified for every Internet Security Association Key Management Protocol (ISAKMP) security association (SA) when the keepalive for that SA is sent based on the existing timer. **Note** A five-percent jitter mechanism value is applied to the timer to avoid security association rekey collisions. If there are many peer routers, and the timer is configured too low, then the router can experience high CPU usage.

Verifying IPsec Configuration

To verify your configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail** **Example:** `Router# show crypto ipsec sa`	Displays the settings used by current SAs.

Configuration Examples for IPsec and NAT

NAT Keepalives Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
 authentication pre-share
```

```
crypto isakmp key 1234 address 10.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set t2
 match address 101
```

Additional References

The following sections provide references related to the IPsec NAT Transparency feature.

Related Documents

Related Topic	Document Title
Additional NAT configuration tasks	• " Configuring NAT for IP Address Conservation " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Using Application Level Gateways with NAT " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Configuring NAT for High Availability " module in the *Cisco IOS XE IP Addressing Services Configuration Guide* • " Integrating NAT with MPLS VPNs " module in the *Cisco IOS XE IP Addressing Services Configuration Guide*
Additional NAT commands	Cisco IOS IP Addressing Services Command Reference
Additional IPsec configuration tasks	" Configuring Security for VPNs with IPsec " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Additional IPsec commands	Cisco IOS Security Command Reference
Information on IKE	" Configuring Internet Key Exchange for IPsec VPNs " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Additional information on IKE dead peer detection	" Easy VPN Server " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*

Standards

Standards	Title
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFCs[22]	Title
RFC 2402	IP Authentication Header
RFC 2406	IP Encapsulating Security Payload (ESP)

[22] Not all supported RFCs are listed.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IPsec NAT Transparency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 245: Feature Information for IPsec NAT Transparency

Feature Name	Releases	Feature Information
IPsec NAT Transparency	Cisco IOS XE Release 2.1	The IPsec NAT Transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPsec. The following commands were introduced or modified: **crypto isamkp nat keepalive**, **access-list (IP extended)**, **show crypto ipsec sa**

Glossary

IKE --Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations (SAs).

IPsec --IP Security. Framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (“peers”), such as Cisco routers.

NAT --Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use on the outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

PAT --Port Address Translation. Like NAT, PAT also translated private IP address to public, routable addresses. Unlike NAT, PAT provides a many-to-one mapping of private addresses to a public address; each instance of the public address is associated with a particular port number to provide uniqueness. PAT can be used in environments where the cost of obtaining a range of public addresses is too expensive for an organization.

CHAPTER 182

IPsec Extended Sequence Number

The Extended Sequence Number (ESN) is an addition to the IPsec standard sequence number that is used to assist high-speed IPsec implementations. IPSec packets have 32 bit sequence numbers, and rekey is mandatory for IKE-keyed IPSec Security Association (SA) after a sequence number rollover. ESN attempts to reduce this high IPsec SA rekey rate by extending the sequence number to 64 bits, this would increase the time before mandatory rekeys.

Prerequisites for IPsec Extended Sequence Number

- ESN must be supported by both IPsec peers involved in establishing a secure connection. This feature will not function if either one of the peers does not support ESN
- Anti-replay configuration is required, when using ESN. For more details see, IPsec Anti-Replay Window Expanding and Disabling.

Restrictions for IPsec Extended Sequence Number

- ESN is only supported on Cisco Catalyst 8500 Series Edge Platforms and Cisco ASR 1000 Series ESP 100-X and ESP 200-X.
- ESN feature is not supported with DES or 3DES algorithms.

Information About IPsec Extended Sequence Number

IPsec Extended Sequence Number

The Extended Sequence Number (ESN) is an addition to the IPsec standard sequence number that is used to assist high-speed IPsec implementations. ESN uses a larger sequence number space than the standard sequence number and it allows the customer to transmit large volumes of data at a high speed without rekeying.

IPSec packets have 32 bit sequence numbers, and rekey is mandatory for IKE-keyed IPSec Security Association (SA) after a sequence number rollover. ESN attempts to reduce this high IPsec SA rekey rate by extending the sequence number to 64 bits, this would increase the time before mandatory rekeys and prevents sequence number rollover. As a result, it lowers the usage of system resources and prevents frequent rekeying on high speed IPsec connections or IPsec implementations that require long IPsec SA lifetime.

How to Configure IPsec Extended Sequence Number

Configuring IPsec Extended Sequence Number

To configure IPsec Extended Sequence Number support, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*]
4. **esn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] **Example:** `Router (config)# crypto ipsec transform-set foo esp-aes esp-sha-hmac`	Configures Transform Sets for IPsec. • There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and the table in "About

	Command or Action	Purpose
		Transform Sets" section provides a list of allowed transform combinations.
Step 4	**esn** **Example:** `Router(cfg-crypto-trans)#[no] esn [optional]`	(Optional) Enables IPsec ESN.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Security Command Reference

Feature Information for IPsec ESN support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 246: Feature Information for IPsec Extended Sequence Number

Feature Name	Releases	Feature Information
IPsec Extended Sequence Number (ESN)	Cisco IOS XE Gibraltar 16.11.1 release	This feature was introduced for the following platforms: • Cisco Catalyst 8500 Series Edge Platforms • Cisco ASR 1000 Series ESP 100-X and ESP 200-X

CHAPTER 183

DF Bit Override Functionality with IPsec Tunnels

The DF Bit Override Functionality with IPsec Tunnels feature allows you to configure the setting of the DF bit when encapsulating tunnel mode IPsec traffic on a global or per-interface level. Thus, if the DF bit is set to clear, routers can fragment packets regardless of the original DF bit setting.

Prerequisites for DF Bit Override Functionality with IPsec Tunnels

IPsec must be enabled on your router.

Restrictions for DF Bit Override Functionality with IPsec Tunnels

Performance Impact

Because each packet is reassembled at the process level, a significant performance impact occurs at a high data rate. Two major caveats are as follows:

- The reassemble queue can fill up and force fragments to be dropped.
- The traffic is slower because of the process switching.

DF Bit Setting Requirement

If several interfaces share the same crypto map using the local address feature, these interfaces must share the same DF bit setting.

Feature Availability

This feature is available only for IPsec tunnel mode. (IPsec transport mode is not affected because it does not provide an encapsulating IP header.)

Information About DF Bit Override Functionality with IPsec Tunnels

Feature Overview

The DF Bit Override Functionality with IPsec Tunnels feature allows you to specify whether your router can clear, set, or copy the Don't Fragment (DF) bit from the encapsulated header. A DF bit is a bit within the IP header that determines whether a router is allowed to fragment a packet.

Some user configurations have hosts that perform the following functions:

- Set the DF bit in packets they send
- Use firewalls that block Internet Control Message Protocol (ICMP) errors from outside the firewall, preventing hosts from learning about the maximum transmission unit (MTU) size outside the firewall
- Use IP Security (IPsec) to encapsulate packets, reducing the available MTU size

If your configurations have hosts that prevent you from learning about the available MTU size, you can configure your router to clear the DF bit and fragment the packet.

Note In compliance with RFC 2401, this feature can be configured globally or per interface. If both levels are configured, the interface configuration will override the global configuration.

How to Configure DF Bit Override Functionality with IPsec Tunnels

Configuring the DF Bit for the Encapsulating Header in Tunnel Mode

To set the DF bit for the encapsulating header in tunnel mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec df-bit** [**clear** | **set** | **copy**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec df-bit** [**clear** | **set** | **copy**] **Example:** `Router (config)# crypto ipsec df-bit set`	Sets the DF bit for the encapsulating header in tunnel mode for all interfaces. To set the DF bit for a specified interface, use the **crypto ipsec df-bit** command in interface configuration mode. **Note** DF bit interface configuration settings override all DF bit global configuration settings.

Verifying DF Bit Setting

To verify the current DF Bit settings on your router, use the **show running-config** command in EXEC mode.

Configuration Examples for DB Bit Override Functionality with IPsec Tunnels

DF Bit Setting Configuration Example

In following example, the router is configured to globally clear the setting for the DF bit and copy the DF bit on the interface named FastEthernet. Thus, all interfaces except FastEthernet will allow the router to send packets larger than the available MTU size; FastEthernet will allow the router to fragment the packet.

```
crypto isakmp policy 1
   hash md5
   authentication pre-share
crypto isakmp key Delaware address 192.168.10.66
crypto isakmp key Key-What-Key address 192.168.11.19
!
!
crypto ipsec transform-set exampleset ah-md5-hmac esp-des
crypto ipsec df-bit clear
!
!
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set exampleset
```

```
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
set transform-set exampleset
match address 102
!
!
interface FastEthernet
   ip address 192.168.10.38 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map armadillo
   crypto ipsec df-bit copy
!
interface FastEthernet1
   ip address 192.168.11.75 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map basilisk
!
interface Serial0
   no ip address
   ip broadcast-address 0.0.0.0
   no ip route-cache
   no ip mroute-cache
```

Additional References

The following sections provide references related to the DF Bit Override Functionality with IPsec Tunnels feature.

Related Documents

Related Topic	Document Title
Internet Key Exchange and IPsec networks	Configuring Internet Key Exchange for IPsec VPNs
IPsec network commands	Cisco IOS Security Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DF Bit Override Functionality with IPsec Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 247: Feature Information for DF Bit Override Functionality with IPsec Tunnels

Feature Name	Releases	Feature Information
DF Bit Override Functionality with IPsec Tunnels	Cisco IOS XE Release 2.1	This feature allows users to specify whether their router can clear, set, or copy the Don't Fragment (DF) bit from the encapsulated header. A DF bit is a bit within the IP header that determines whether a router is allowed to fragment a packet. The following commands were introduced or modified: **crypto ipsec df-bit**.

CHAPTER 184

IPsec Security Association Idle Timers

When a router running the Cisco IOS XE software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers. The IPsec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. Benefits of this feature include:

- Increased availability of resources
- Improved scalability of Cisco IOS XE IPsec deployments. Because this feature prevents the wasting of resources by idle peers, more resources will be available to create new SAs as required.

Prerequisites for IPsec Security Association Idle Timers

You must configure Internet Key Exchange (IKE) as described in the " Configuring Internet Key Exchange Security Protocol " chapter of the *Cisco IOS XE Security Configuration Guide* .

Information About IPsec Security Association Idle Timers

Lifetimes for IPsec Security Associations

The Cisco IOS software currently allows the configuration of lifetimes for IPsec SAs. Lifetimes can be configured globally or per crypto map. There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. A security association expires after the first of these lifetimes is reached.

IPsec Security Association Idle Timers

The IPsec SA idle timers are different from the global lifetimes for IPsec SAs. The expiration of the global lifetime is independent of peer activity. The IPsec SA idle timer allows SAs associated with inactive peers to be deleted before the global lifetime has expired.

If the IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

Note If the last IPsec SA to a given peer is deleted due to idle timer expiration, the Internet Key Exchange (IKE) SA to that peer will also be deleted.

How to Configure IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally

This task configures the IPsec SA idle timer globally. The idle timer configuration will be applied to all SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association idle-time** *seconds* **Example:** `Router(config)# crypto ipsec security-association idle-time 600`	Configures the IPsec SA idle timer. • The *seconds* argument specifies the time, in seconds, that the idle timer will allow an inactive peer to maintain an SA. Valid values for the *seconds* argument range from 60 to 86400.

Configuring the IPsec SA Idle Timer per Crypto Map

This task configures the IPsec SA idle timer for a specified crypto map. The idle timer configuration will be applied to all SAs under the specified crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-number* **ipsec-isakmp**
4. **set security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-number* **ipsec-isakmp** **Example:** `Router(config)# crypto map test 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**set security-association idle-time** *seconds* **Example:** `Router(config-crypto-map)# set security-association idle-time 600`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used. • The *seconds* argument is the number of seconds for which the current peer can be idle before the default peer is used. Valid values are 60 to 86400.

Configuration Examples for IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally Example

The following example globally configures the IPsec SA idle timer to drop SAs for inactive peers after 600 seconds:

```
crypto ipsec security-association idle-time 600
```

Configuring the IPsec SA Idle Timer per Crypto Map Example

The following example configures the IPsec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

```
crypto map test 1 ipsec-isakmp
 set security-association idle-time 600
```

Additional References

The following sections provide references related to the IPsec Security Association Idle Timers feature.

Related Documents

Related Topic	Document Title
Additional information about configuring IKE	Internet Key Exchange for IPsec VPNs
Additional information about configuring global lifetimes for IPsec SAs	• Configuring Security for VPNs with IPsec • IPsec Preferred Peer
Additional Security commands	Cisco IOS Security Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	---

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IPsec Security Association Idle Timers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 248: Feature Information for IPsec Security Association Idle Timers

Feature Name	Releases	Feature Information
IPsec Security Association Idle Timers	Cisco IOS XE Release 2.1	When a router running the Cisco IOS XE software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers. The IPsec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. The following command was introduced or modified: **crypto ipsec security-association idle-time**.
	Cisco IOS XE Release 2.1	The **set security-association idle-time** command was added, allowing for the configuration of an IPsec idle timer for a specified crypto map. The following command was introduced or modified: **set security-association idle-time**.

CHAPTER 185

IPv6 IPsec Quality of Service

The IPv6 IPsec QoS feature allows the quality of service (QoS) policies to be applied to IPv6 IPsec.

Information About IPv6 IPsec QoS

IPv6 IPsec QoS Overview

The IPv6 IPsec QoS feature applies the quality of service (QoS) policies to IPV6 IPsec. This feature supports the following functionalities:

- Crypto LLQ QoS—Traffic that is classified by QoS and marked as priority level 1 or 2 by traditional Cisco Modular QoS CLI (MQC) QoS configuration, for example PAK priority, is enqueued to the priority queue before the crypto processor. The low latency queuing (LLQ) for IPsec encryption engines helps reduce packet latency for priority traffic.
- IPsec QoS Pre-Classify—QoS pre-classify is configured under a crypto map to enable IPsec to save the original Layer 3 and Layer 4 header before the encryption so that QoS can do the classification using the saved header.
- QoS group-based LLQ—The QoS group-based LLQ feature allows IPsec to check the LLQ QoS group setting to determine whether a packet is a high priority packet before it is enqueued to low latency queuing (LLQ).

How to Configure IPv6 IPsec QoS

Configuring Crypto LLQ QoS

When IPsec and QoS are configured on a physical interface and if the QoS policy has priority class, IPSec will classify the packet based on the policy attached to the interface. It will enqueue the packet matching

priority class into Low Latency Queue. The high-priority packet will be enqueued to low latency queueing (LLQ).

Perform this task to attach a service policy to the output interface and enable LLQ for IPsec encryption engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *physical-interface-name*
4. **ipv6 address** *{ipv6-address /prefix-length | prefix-name sub-bits/prefix-length}*
5. **service-policy output** *policy-map*
6. **ipv6 crypto map** *map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *physical-interface-name* **Example:** `Device(config)# interface GigabitEthernet0/0/1`	Specifies the interface using the LLQ for IPsec encryption engines.
Step 4	**ipv6 address** *{ipv6-address /prefix-length | prefix-name sub-bits/prefix-length}* **Example:** `Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64`	Configures an IPv6 address on an interface.
Step 5	**service-policy output** *policy-map* **Example:** `Device(config-if)# service-policy output p1`	Attaches the specified service policy map to the output interface and enables LLQ for IPsec encryption engines.
Step 6	**ipv6 crypto map** *map-name* **Example:** `Device(config-if)# ipv6 crypto map CMAP_1`	Enables an IPv6 crypto map on an interface.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring QoS Pre-classify

Configuring Pre-classify on the Crypto Map

The **qos pre-classify** command is applied on the crypto map, allowing configuration on a per-tunnel basis. QoS policy is applied to Packets based on the L3 and L4 Header before encryption.

Perform this task to apply the QoS pre-classify on the crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 crypto map** *map-name*
4. **qos pre-classify**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 crypto map** *map-name* **Example:** `Device(config-if)# ipv6 crypto map CM_V6`	Enters crypto map configuration mode and specifies the crypto map to be configured.
Step 4	**qos pre-classify** **Example:** `Device(config-if)# qos pre-classify`	Enables QoS pre-classify on the crypto map.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Pre-classify on the Tunnel Interface

The **qos pre-classify** command is applied on the IPv6 IPsec tunnel interface, making QoS a configuration option on a per-tunnel basis.

Perform this task to apply the QOS pre-classify on the tunnel interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel-interface-name*
4. **ipv6 address** *{ipv6-address /prefix-length | prefix-name sub-bits/prefix-length}*
5. **qos pre-classify**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel-interface-name* **Example:** `Device(config)# interface Tunnel1`	Enters interface configuration mode and specifies the tunnel or virtual interface to configure.
Step 4	**ipv6 address** *{ipv6-address /prefix-length | prefix-name sub-bits/prefix-length}* **Example:** `Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64`	Configures an IPv6 address on an interface.
Step 5	**qos pre-classify** **Example:**	Enables QoS pre-classify on the tunnel interface.

	Command or Action	Purpose
	`Device(config-if)# qos pre-classify`	
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring LLQ QoS Group

The **platform ipsec llq qos-group** command enables low latency queuing for traffic that matches the QoS groups configured with this command.

Perform this task to enable LLQ for QoS groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform ipsec llq qos-group** *group-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**platform ipsec llq qos-group** *group-number* **Example:** `Device(config)# platform ipsec llq qos-group 1`	Specifies the QoS group to enable LLQ. Valid values are from 1 to 99.
Step 4	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for QoS

Example: Configuring Crypto LLQ QoS

The following example shows how to specify the service policy map to the output interface and enable an IPv6 crypto map on an interface.

```
!
class-map match-all c2
  match precedence 5  6  7
class-map match-all c1
  match precedence 0  1  2  3

policy-map p1
  class c1
    priority percent 10
  class c2
    bandwidth remaining percent 3

crypto map ipv6 CMAP_1 1 ipsec-isakmp
  set peer address 2001:DB8:FFFF::1
  set transform-set ESP-3DES-SHA
  match address 102

interface GigabitEthernet0/0/1
  ipv6 address 2001:DB8:FFFF::2/64
  ipv6 crypto map CMAP_1
  service-policy output p1
```

Example: Configuring Pre-classify on the Crypto Map

The following example shows how to enable QoS pre-classification using the **qos pre-classify** command on the crypto map CM_V6.

```
!
crypto map ipv6 CM_V6 10 ipsec-isakmp
  match address ACL_IPV6_1
  set transform-set set1
  set peer 2001:DB8:FFFF::1
  qos pre-classify
!
interface GigabitEthernet0/0/1
  ipv6 address 2001:DB8:FFFF::2/64
  service-policy output policy1
  ipv6 crypto map CM_V6
```

Example: Configuring Pre-classify on the Tunnel Interface

The following example shows how to enable QoS pre-classification using the **qos pre-classify** command on the tunnel interface tunnel1.

```
interface GigabitEthernet1/1/2
  ipv6 address 2001:DB8:1::F/64
  service-policy output policy1
!
interface Tunnel1
  ipv6 address 2001:DB8:2::F/64
  qos pre-classify
  ipv6 mtu 1400
  tunnel protection ipsec profile greprof
```

Example: Configuring LLQ QoS Group

The following example shows how to configure low latency queuing on a QoS group.

```
!
platform ipsec llq qos-group 1
platform ipsec llq qos-group 49
!
!
crypto map ipv6 cmap 1 ipsec-isakmp
  set peer 2001:DB8:FFFF:1::E/64
  set security-association lifetime seconds 600
  set transform-set aes-192
  match address 102
!
!
class-map match-all c1
  match precedence 5
class-map match-all c2
  match precedence 2
class-map match-all c3
  match precedence 4
class-map match-all c4
  match precedence 3
!
policy-map p1
  class c3
    set qos-group 20
  class c1
    set qos-group 49
  class c4
    set qos-group 77
!
policy-map p2
  class class-default
    set qos-group 1
!
interface GigabitEthernet0/2/0
  ipv6 address
  negotiation auto
  cdp enable
  ipv6 crypto map cmap
  service-policy input p2
!
!
interface GigabitEthernet0/2/7
  ipv6 address 2001:DB8:FFFF:1::F/64
  negotiation auto
  cdp enable
```

```
  service-policy input p1
!
```

Additional References for IPv6 IPsec QoS

Related Documents

Related Topic	Document Title
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
IPv6 Commands	*IPv6 Command Reference*
QoS Commands	*Cisco IOS Quality of Service Solutions Command Reference*
IPv6 Addressing and Connectivity	*IPv6 Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 IPsec QoS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 249: Feature Information for IPv6 IPsec QoS

Feature Name	Releases	Feature Information
IPv6 IPsec QoS	15.4(1)S	The IPv6 IPsec QoS feature allows the QoS policies to be applied to IPv6 IPsec. This feature supports the following functionalities: • Crypto LLQ QoS • IPsec QoS Pre-Classify • QoS group-based LLQ The following command was modified: **ipv6 crypto map**

CHAPTER 186

IPv6 Virtual Tunnel Interface

Cisco IOS IPv6 security features for your Cisco networking devices can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Cisco IOS IPsec functionality provides network data encryption at the IP packet level, offering robust, standards-based security. IPsec provides data authentication and antireplay services in addition to data confidentiality services.

IPsec is a mandatory component of IPv6 specification. IPv6 IPsec tunnel mode and encapsulation is used to protect IPv6 unicast and multicast traffic. This document provides information about implementing IPsec in IPv6 security.

Information About IPv6 Virtual Tunnel Interface

IPsec for IPv6

IP Security, or IPsec, is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers. IPsec provides the following optional network security services. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality--The IPsec sender can encrypt packets before sending them across a network.
- Data integrity--The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data origin authentication--The IPsec receiver can authenticate the source of the IPsec packets sent. This service depends upon the data integrity service.
- Antireplay--The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be sent across a public network without observation, modification, or spoofing. IPsec functionality is similar in both IPv6 and IPv4; however, site-to-site tunnel mode only is supported in IPv6.

In IPv6, IPsec is implemented using the AH authentication header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.

The Internet Key Exchange (IKE) protocol is a key management protocol standard that is used in conjunction with IPsec. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

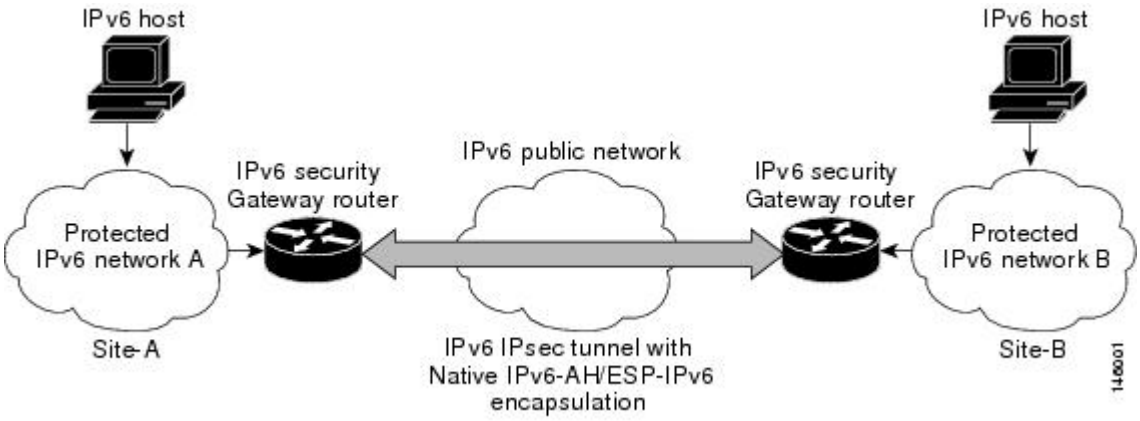
IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE) (see the figure below). This functionality is similar to the security gateway model using IPv4 IPsec protection.

IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface

The IPsec virtual tunnel interface (VTI) provides site-to-site IPv6 crypto protection of IPv6 traffic. Native IPv6 IPsec encapsulation is used to protect all types of IPv6 unicast and multicast traffic.

The IPsec VTI allows IPv6 routers to work as security gateways, establish IPsec tunnels between other security gateway routers, and provide crypto IPsec protection for traffic from internal networks when it is sent across the public IPv6 Internet (see the figure below). This functionality is similar to the security gateway model using IPv4 IPsec protection.

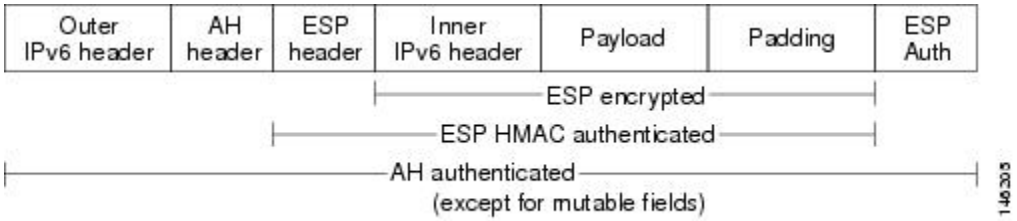
Figure 94: IPsec Tunnel Interface for IPv6



When the IPsec tunnel is configured, IKE and IPsec security associations (SAs) are negotiated and set up before the line protocol for the tunnel interface is changed to the UP state. The remote IKE peer is the same as the tunnel destination address; the local IKE peer will be the address picked from tunnel source interface which has the same IPv6 address scope as tunnel destination address.

The following figures shows the IPsec packet format.

Figure 95: IPv6 IPsec Packet Format



How to Configure IPv6 Virtual Tunnel Interface

Configuring a VTI for Site-to-Site IPv6 IPsec Protection

Defining an IKE Policy and a Preshared Key in IPv6

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer--each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Note If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime that is less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime--from the remote peer's policy--will be used.)

If a match is found, IKE will complete negotiation, and IPsec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPsec will not be established.

Note Depending on which authentication method is specified in a policy, additional configuration might be required. If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPsec SAs, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IPv6 address, depending on how you have set the ISAKMP identity of the router.

By default, a peer's ISAKMP identity is the IPv6 address of the peer. If appropriate, you could change the identity to be the peer's hostname instead. As a general rule, set the identities of all peers the same way--either all peers should use their IPv6 addresses or all peers should use their hostnames. If some peers use their hostnames and some peers use their IPv6 addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a DNS lookup is unable to resolve the identity.

Perform this task to create an IKE policy and a preshared key in IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
5. **hash** {**sha** | **md5**}
6. **group** {**1** | **2** | **5**}
7. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
8. **lifetime** *seconds*
9. **exit**
10. **crypto isakmp key** password-type keystring *keystring* { **address** *peer-address*| **ipv6** {*ipv6-address* / *ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
11. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
12. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}} **key** *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 15`	Defines an IKE policy, and enters ISAKMP policy configuration mode. • Policy number 1 indicates the policy with the highest priority. The smaller the *priority* argument value, the higher the priority.
Step 4	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Example:** `Router(config-isakmp-policy)# authentication pre-share`	Specifies the authentication method within an IKE policy. • The **rsa-sig** and **rsa-encr** keywords are not supported in IPv6.
Step 5	**hash** {**sha** | **md5**} **Example:** `Router(config-isakmp-policy)# hash md5`	Specifies the hash algorithm within an IKE policy.
Step 6	**group** {**1** | **2** | **5**} **Example:** `Router(config-isakmp-policy)# group 2`	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 7	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**} **Example:** `Router(config-isakmp-policy)# encryption 3des`	Specifies the encryption algorithm within an IKE policy.
Step 8	**lifetime** *seconds* **Example:** `Router(config-isakmp-policy)# lifetime 43200`	Specifies the lifetime of an IKE SA. • Setting the IKE lifetime value is optional.
Step 9	**exit** **Example:** `Router(config-isakmp-policy)# exit`	Exits ISAKMP policy configuration mode and enter global configuration mode.
Step 10	**crypto isakmp key** password-type keystring *keystring* { **address** *peer-address*| **ipv6** {*ipv6-address* / *ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**] **Example:** `Router(config)# crypto isakmp key 0 my-preshare-key-0 address ipv6 3ffe:1001::2/128`	Configures a preshared authentication key.
Step 11	**crypto keyring** *keyring-name* [**vrf** *fvrf-name*] **Example:**	Defines a crypto keyring to be used during IKE authentication and enters config-keyring mode.

	Command or Action	Purpose
	`Router(config)# crypto keyring keyring1`	
Step 12	**pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}} **key** *key* **Example:** `Router (config-keyring)# pre-shared-key ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Defines a preshared key to be used for IKE authentication.

Configuring ISAKMP Aggressive Mode

You likely do not need to configure aggressive mode in a site-to-site scenario. The default mode is typically used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**address** {*ipv4-address* | **ipv6** *ipv6-address ipv6-prefix-length*} | **hostname** *fqdn-hostname*}
4. **set aggressive-mode client-endpoint** {*client-endpoint* | **ipv6** *ipv6-address*}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp peer** {**address** {*ipv4-address* | **ipv6** *ipv6-address ipv6-prefix-length*} | **hostname** *fqdn-hostname*} **Example:** `Router(config)# crypto isakmp peer address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Enables an IPsec peer for IKE querying for tunnel attributes.
Step 4	**set aggressive-mode client-endpoint** {*client-endpoint* | **ipv6** *ipv6-address*} **Example:**	Defines the remote peer's IPv6 address, which will be used by aggressive mode negotiation. The remote peer's address is usually the client side's end-point address.

	Command or Action	Purpose
	`Router(config-isakmp-peer)# set aggressive mode client-endpoint ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	
Step 5	**end** **Example:** `Router(config-isakmp-peer)# end`	Exits crypto ISAKMP peer configuration mode and returns to privileged EXEC mode.

Defining an IPsec Transform Set and IPsec Profile

Perform this task to define an IPsec transform set. A transform set is a combination of security protocols and algorithms that is acceptable to the IPsec routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
4. **crypto ipsec profile** *name*
5. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des`	Defines a transform set, and places the router in crypto transform configuration mode.
Step 4	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile profile0`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.

	Command or Action	Purpose
Step 5	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6* **Example:** `Router (config-crypto-transform)# set-transform-set myset0`	Specifies which transform sets can be used with the crypto map entry.

Defining an ISAKMP Profile in IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name* [**accounting** *aaalist*
4. **self-identity** {**address** | **address ipv6**] | **fqdn** | **user-fqdn** *user-fqdn*}
5. **match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrf*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* [**accounting** *aaalist* **Example:** `Router(config)# crypto isakmp profile profile1`	Defines an ISAKMP profile and audits IPsec user sessions.
Step 4	**self-identity** {**address**	**address ipv6**]	**fqdn**	**user-fqdn** *user-fqdn*} **Example:** `Router(config-isakmp-profile)# self-identity address ipv6`	Defines the identity that the local IKE uses to identify itself to the remote peer.

	Command or Action	Purpose
Step 5	**match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrf*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*} **Example:** `Router(config-isakmp-profile)# match identity address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128`	Matches an identity from a remote peer in an ISAKMP profile.
Step 6	**end** **Example:** `Router(config-isakmp-profile)# end`	Exits ISAKMP profile configuration mode and returns to privileged EXEC mode.

Configuring IPv6 IPsec VTI

Before you begin

Use the **ipv6 unicast-routing** command to enable IPv6 unicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface tunnel** *tunnel-number*
5. **ipv6 address** *ipv6-address/prefix*
6. **ipv6 enable**
7. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
8. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
9. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}
10. **tunnel protection ipsec profile** *name* [**shared**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables IPv6 unicast routing. You only need to enable IPv6 unicast routing once, not matter how many interface tunnels you want to configure.
Step 4	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 5	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 3FFE:C000:0:7::/64 eui-64`	Provides an IPv6 address to this tunnel interface, so that IPv6 traffic can be routed to this tunnel.
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 on this tunnel interface.
Step 7	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source ethernet0`	Sets the source address for a tunnel interface.
Step 8	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 2001:DB8:1111:2222::1`	Specifies the destination for a tunnel interface.
Step 9	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Example:** `Router(config-if)# tunnel mode ipsec ipv6`	Sets the encapsulation mode for the tunnel interface. For IPsec, only the **ipsec ipv6** keywords are supported.
Step 10	**tunnel protection ipsec profile** *name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile profile1`	Associates a tunnel interface with an IPsec profile. IPv6 does not support the **shared** keyword.

	Command or Action	Purpose
Step 11	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying IPsec Tunnel Mode Configuration

SUMMARY STEPS

1. **show adjacency** [**summary** [*interface-type interface-number*]] | [**prefix**] [**interface** *interface-number*] [**connectionid** *id*] [**link** {**ipv4**| **ipv6** | **mpls**}] [**detail**]
2. **show crypto engine** {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**}
3. **show crypto ipsec sa** [**ipv6**] [*interface-type interface-number*] [**detailed**]
4. **show crypto isakmp peer** [**config** | **detail**]
5. **show crypto isakmp policy**
6. **show crypto isakmp profile** [**tag** *profilename* | **vrf** *vrfname*]
7. **show crypto map** [**interface** *interface* | **tag** *map-name*]
8. **show crypto session** [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port** *remote-port*]] | **detail**] | **fvfr** *vrf-name* | **ivrf** *vrf-name*]
9. **show crypto socket**
10. **show ipv6 access-list** [*access-list-name*]
11. **show ipv6 cef** [*ipv6-prefix / prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]]
12. **show interface** *type number* **stats**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show adjacency** [**summary** [*interface-type interface-number*]]	[**prefix**] [**interface** *interface-number*] [**connectionid** *id*] [**link** {**ipv4**	**ipv6**	**mpls**}] [**detail**] **Example:** `Router# show adjacency detail`	Displays information about the Cisco Express Forwarding adjacency table or the hardware Layer 3-switching adjacency table.
Step 2	**show crypto engine** {**accelerator**	**brief**	**configuration**	**connections** [**active**	**dh**	**dropped-packet**	**show**]	**qos**} **Example:** `Router# show crypto engine connection active`	Displays a summary of the configuration information for the crypto engines.
Step 3	**show crypto ipsec sa** [**ipv6**] [*interface-type interface-number*] [**detailed**]	Displays the settings used by current SAs in IPv6.

	Command or Action	Purpose
	Example: `Router# show crypto ipsec sa ipv6`	
Step 4	**show crypto isakmp peer** [**config** | **detail**] **Example:** `Router# show crypto isakmp peer detail`	Displays peer descriptions.
Step 5	**show crypto isakmp policy** **Example:** `Router# show crypto isakmp policy`	Displays the parameters for each IKE policy.
Step 6	**show crypto isakmp profile** [**tag** *profilename* | **vrf** *vrfname*] **Example:** `Router# show crypto isakmp profile`	Lists all the ISAKMP profiles that are defined on a router.
Step 7	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays the crypto map configuration. The crypto maps shown in this command output are dynamically generated. The user does not have to configure crypto maps.
Step 8	**show crypto session** [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port** *remote-port*]] | **detail**] | **fvfr** *vrf-name* | **ivrf** *vrf-name*] **Example:** `Router# show crypto session`	Displays status information for active crypto sessions. IPv6 does not support the **fvfr** or **ivrf** keywords or the *vrf-name* argument.
Step 9	**show crypto socket** **Example:** `Router# show crypto socket`	Lists crypto sockets.
Step 10	**show ipv6 access-list** [*access-list-name*] **Example:** `Router# show ipv6 access-list`	Displays the contents of all current IPv6 access lists.
Step 11	**show ipv6 cef** [*ipv6-prefix / prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]] **Example:** `Router# show ipv6 cef`	Displays entries in the IPv6 Forwarding Information Base (FIB).

	Command or Action	Purpose
Step 12	**show interface** *type number* **stats** **Example:** `Router# show interface fddi 3/0/0 stats`	Displays numbers of packets that were process switched, fast switched, and distributed switched.

Troubleshooting IPsec for IPv6 Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto engine packet** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IPsec network events.
Step 3	**debug crypto engine packet** [**detail**] **Example:** `Router# debug crypto engine packet`	Displays the contents of IPv6 packets. **Caution** Using this command could flood the system and increase CPU usage if several packets are being encrypted.

Configuration Examples for IPv6 Virtual Tunnel Interface

Example: Configuring a VTI for Site-to-Site IPv6 IPsec Protection

```
crypto isakmp policy 1
  encryption aes
  authentication pre-share
  group 14
!
crypto isakmp key myPreshareKey0 address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128
crypto isakmp keepalive 30 30
!
```

```
crypto ipsec transform-set Trans1 ah-sha-hmac esp-aes
!
crypto ipsec profile profile0
  set transform-set Trans1
!
ipv6 cef
!
interface Tunnel0
  ipv6 address 3FFE:1001::/64 eui-64
  ipv6 enable
  ipv6 cef
  tunnel source Ethernet2/0
  tunnel destination 3FFE:2002::A8BB:CCFF:FE01:2C02
  tunnel mode ipsec ipv6
  tunnel protection ipsec profile profile0
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	Cisco IOS Security Command Reference
QoS Commands	Cisco IOS Quality of Service Solutions Command Reference
Weighted Fair Queueing	Configuring Weighted Fair Queueing feature module.

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Virtual Tunnel Interface

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 250: Feature Information for IPv6 Virtual Tunnel Interface

Feature Name	Releases	Feature Information
IPv6 Virtual Tunnel Interface	Cisco IOS XE Release 2.4	IPsec is a framework of open standards that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers. The following commands were introduced or modified: **authentication (IKE policy)**, **crypto ipsec profile**, **crypto isakmp key**, **crypto isakmp peer**, **crypto isakmp policy**, **crypto isakmp profile**, **crypto keyring**, **debug crypto ipv6 ipsec**, **encryption (IKE policy)**, **group (IKE policy)**, **hash (IKE policy)**, **lifetime (IKE policy)**, **match identity**, **pre-shared-key**, **self-identity**, **set aggressive-mode client-endpoint**, **set transform-set**, **show adjacency**, **show crypto engine**, **show crypto ipsec sa**, **show crypto isakmp peers**, **show crypto isakmp policy**, **show crypto isakmp profile**, **show crypto map**, **show crypto session**, **show crypto socket**, **show ipv6 access-list**, **show ipv6 cef**, **tunnel destination**, **tunnel mode**, **tunnel source**.

PART XVIII

IPsec Management Plane

CHAPTER 187

IP Security VPN Monitoring

The IP Security VPN Monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. Session monitoring enhancements include the following:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file
- Summary listing of crypto session status
- Syslog notification for crypto session up or down status
- Ability to clear both IKE and IP Security (IPsec) security associations (SAs) using one command-line interface (CLI)

Prerequisites for IP Security VPN Monitoring

- You should be familiar with IPSec and encryption.
- Your router must support IPSec, and before using the IP Security VPN Monitoring feature, you must have configured IPSec on your router.

Restrictions for IP Security VPN Monitoring

- You must be running Cisco IOS XE k8 or k9 crypto images on your router.

Information About IPsec VPN Monitoring

Background Crypto Sessions

A crypto session is a set of IPSec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPSec security associations (for data traffic--one per each direction). There may be duplicated IKE security associations (SAs) and IPSec SAs or duplicated IKE SAs or IPSec SAs for the same session in the duration of rekeying or because of simultaneous setup requests from both sides.

Per-IKE Peer Description

The Per-IKE Peer Description function allows you to enter a description of your choosing for an IKE peer. The unique peer description, which can include up to 80 characters, can be used whenever you are referencing that particular IKE peer. To add the peer description, use the **description** command.

Note IKE peers that "sit" behind a Network Address Translation (NAT) device cannot be uniquely identified; therefore, they have to share the same peer description.

The primary application of this description field is for monitoring purposes (for example, when using **show** commands or for logging [syslog messages]). The description field is purely informational (for example, it cannot act as a substitute for the peer address or FQDN when defining crypto maps).

Summary Listing of Crypto Session Status

You can get a list of all the active VPN sessions by entering the **show crypto session** command. The listing will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by whom the IPSec SAs are created
- IPSec SAs serving the flows of a session

Multiple IKE or IPSec SAs may be established for the same peer (for the same session), in which case IKE peer descriptions will be repeated with different values for the IKE SAs that are associated with the peer and for the IPSec SAs that are serving the flows of the session.

You can also use the **show crypto session detail** variant of this command to obtain more detailed information about the sessions.

Syslog Notification for Crypto Session Up or Down Status

The Syslog Notification for Crypto Session Up or Down Status function provides syslog notification every time the crypto session comes up or goes down.

The following is a sample syslog notification showing that a crypto session is up:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

The following is a sample syslog notification showing that a crypto session is down:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

IKE and IPsec Security Exchange Clear Command

The **clear crypto session** command allows you to clear both IKE and IPsec with a single command. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you need to provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address will be used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you use the **clear crypto session** command, all the sessions (and their IKE SAs and IPsec SAs) that share the IP address as a local crypto endpoint (IKE local address) will be cleared. If you do not provide a parameter when you use the **clear crypto session** command, all IPsec SAs and IKE SAs that are in the router will be deleted.

How to Configure IP Security VPN Monitoring

Adding the Description of an IKE Peer

To add the description of an IKE peer to an IPsec VPN session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**ip-address** *ip-address*}
4. **description**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp peer** {**ip-address** *ip-address*} **Example:** `Router (config)# crypto isakmp peer address 10.2.2.9`	Enables an IPsec peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and enters ISAKMP peer configuration mode.
Step 4	**description** **Example:** `Router (config-isakmp-peer)# description connection from site A`	Adds a description for an IKE peer.

Verifying Peer Descriptions

To verify peer descriptions, use the **show crypto isakmp peer** command.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp peer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp peer** **Example:** `Router# show crypto isakmp peer`	Displays peer descriptions.

Examples

The following output example verifies that the description “connection from site A” has been added for IKE peer 10.2.2.9:

```
Router# show crypto isakmp peer
Peer: 10.2.2.9 Port: 500
```

```
 Description: connection from site A
  flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

The following output example verifies that the description “connection from site A” has been added for IKE peer 10.2.2.9:

```
Router# show crypto isakmp peer
Peer: 10.2.2.9 Port: 500
 Description: connection from site A
  flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

Clearing a Crypto Session

To clear a crypto session, use the **clear crypto session** command from the router command line. No configuration statements are required in the configuration file to use this command.

SUMMARY STEPS

1. **enable**
2. **clear crypto session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPSec and IKE SAs).

Configuration Examples for IP Security VPN Monitoring

show crypto session Command Output Examples

The following is sample output for the **show crypto session** output without the **detail** keyword:

```
Router# show crypto session
Crypto session current status
Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
   IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
   IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
         Active SAs: 2, origin: crypto map
```

The following is sample output using the **show crypto session command and the detail** keyword:

```
Router# show crypto session detail
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
      Desc: this is my peer at 10.1.1.3:500 Green
      Phase1_id: 10.1.1.3
  IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
          Capabilities:(none) connid:3 lifetime:22:03:24
  IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
        Active SAs: 0, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
  IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
        Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

Additional References

The following sections provide references related to IP Security VPN Monitoring.

Related Documents

Related Topic	Document Title
IP security, encryption, and IKE	• Configuring Internet Key Exchange for IPsec VPNs • Configuring Security for VPNs with IPsec
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for exiting standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for exiting MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for exiting RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IP Security VPN Monitoring

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 251: Feature Information for IP Security VPN Monitoring

Feature Name	Releases	Feature Information
IP Security VPN Monitoring	Cisco IOS XE Release 2.1	The IP Security VPN Monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the VPN and monitor the end-user interface. Session monitoring enhancements include the following: • Ability to specify an IKE peer description in the configuration file • Summary listing of crypto session status • Syslog notification for crypto session up or down status Ability to clear both IKE and IPsec SAs using one CLI • The following commands were introduced or modified: **clear crypto session**, **description (isakmp peer)**, **show crypto isakmp peer**, **show crypto session**.

CHAPTER 188

IPsec and IKE MIB Support forCisco VRF-Aware IPsec

The IPsec and IKE MIB Support for the Virtual Private Network routing and forwarding- (VRF-) aware IP security (IPsec) feature allows VRF-aware IPsec to be managed with MIBs, which provide the details of IPsec statistics and performance metrics on a per VRF basis.

Prerequisites for IPsec and IKE MIB Support forCisco VRF-Aware IPsec

- You should be familiar with configuring Simple Network Management Protocol (SNMP).

Information About IPsec and IKE MIB Support forCisco VRF-Aware IPsec

MIBs Supported by the IPsec and IKE MIB Support forCisco VRF-Aware IPsec Feature

- CISCO-IPSEC-FLOW-MONITOR-MIB supports IKE and IPSEC per-tunnel history and failure information. The length of this history and failure information can be configured and must be maintained on a per-VRF basis. The table sizes are controlled by using the **crypto mib ipsec flowmib history tunnel size number** and **crypto mib ipsec flowmib history failure size** commands in global configuration mode.

- CISCO-IPSEC-MIB
- CISCO-IPSEC-POLICY-MAP-MIB is supported. However, because this MIB applies to the entire router rather than to a specific VPN VRF instance, it is not VRF aware; therefore, polling of the object identifiers (OIDs) that belong to this MIB is accomplished with respect to the global VRF context.

SNMP Traps Supported by the IPsec and IKE MIB Support for Cisco VRF-Aware IPsec Feature

The following IKE and IPsec tunnel start and stop traps must go with their corresponding VRF:

- IPSEC_TUNNEL_STOP
- IKE_TUNNEL_STOP
- IPSEC_TUNNEL_START
- IKE_TUNNEL_START

The following traps are global traps that have been modified for the Cisco VRF-Aware IPsec feature:

- TOO_MANY_SAS_CREATED
- CRYPTOMAP_ADDED
- CRYPTOMAPSET_ATTACHED
- CRYPTOMAP_DELETED
- CRYPTOMAPSET_DELETED
- ISAKMP_POLICY_ADDED
- ISAKMP_POLICY_DELETED

How to Configure IPsec and IKE MIB Support for Cisco VRF-Aware IPsec

No special configuration is needed for this feature. The SNMP framework can be used to manage VRF-aware IPsec using MIBs. See the Configuration Examples for IPsec and IKE MIB Support for Cisco VRF-Aware IPsec section for more information.

The following section provides information about troubleshooting this feature:

How to Troubleshoot the IPsec and IKE MIB Support for Cisco VRF-Aware IPsec Feature

The following **debug crypto mib** command and keywords may be used to display information about the IPsec and Internet Key Exchange (IKE) MIB as it relates to Cisco VRF-aware IPsec.

SUMMARY STEPS

1. **enable**
2. **debug crypto mib detail**
3. **debug crypto mib error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto mib detail** **Example:** `Router# debug crypto mib detail`	Displays different events as they occur in the IPsec MIB subsystem. • Due consideration should be given to enabling **debug crypto mib detail**because the output for the **detail** keyword can be quite long.
Step 3	**debug crypto mib error** **Example:** `Router# debug crypto mib error`	Displays error events in the MIB agent.

Configuration Example for IPsec and IKE MIB Support for Cisco VRF-Aware IPsec

Configuration That Has Two VRFs Examples

The following output example is for a typical hub configuration that has two VRFs. The output is what you would see if you were to poll for the IPsec security association (SA). Router 3745b is the VRF-aware router.

Two VRFs Configured

The following output shows that two VRFs have been configured (vrf1 and vrf2).

```
Router3745b# show running-config
Building configuration...
Current configuration : 6567 bytes
!
version 12.4
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
!
hostname ipsecf-3745b
```

```
!
boot-start-marker
boot-end-marker
!
no logging console
enable password lab
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip cef
!
!
ip vrf vrf1
 rd 1:101
 context vrf-vrf1-context
 route-target export 1:101
 route-target import 1:101
!
ip vrf vrf2
 rd 2:101
 context vrf-vrf2-context
 route-target export 2:101
 route-target import 2:101
!
no ip domain lookup
!
!
crypto keyring vrf1-1 vrf vrf1
  pre-shared-key address 10.1.1.1 255.255.255.0 key vrf1-1
crypto keyring vrf2-1 vrf vrf2
  pre-shared-key address 10.1.2.1 255.255.255.0 key vrf2-1
!
!
crypto isakmp policy 1
 authentication pre-share
!
crypto isakmp policy 50
 authentication pre-share
crypto isakmp key global1-1 address 10.1.151.1
crypto isakmp key global2-1 address 10.1.152.1
crypto isakmp profile vrf1-1
   keyring vrf1-1
   match identity address 10.1.1.1 255.255.255.255 vrf1
crypto isakmp profile vrf2-1
   keyring vrf2-1
   match identity address 10.1.2.1 255.255.255.255 vrf2
!
crypto ipsec security-association lifetime kilobytes 99000
crypto ipsec security-association lifetime seconds 5000
!
crypto ipsec transform-set tset ah-sha-hmac esp-des esp-sha-hmac
!
crypto map global1-1 10 ipsec-isakmp
 set peer 10.1.151.1
 set transform-set tset
 match address 151
!
crypto map global2-1 10 ipsec-isakmp
```

```
 set peer 10.1.152.1
 set transform-set tset
 match address 152
!
crypto map vrf1-1 10 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set tset
 set isakmp-profile vrf1-1
 match address 101
!
crypto map vrf2-1 10 ipsec-isakmp
 set peer 10.1.2.1
 set transform-set tset
 set isakmp-profile vrf2-1
 match address 102
!
!
interface FastEthernet0/0
 ip address 10.1.38.25 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
!
interface Serial0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface FastEthernet0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial1/0
 no ip address
 encapsulation frame-relay
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 no keepalive
 serial restart-delay 0
 clock rate 128000
 no frame-relay inverse-arp
!
interface Serial1/0.1 point-to-point
 ip vrf forwarding vrf1
 ip address 10.3.1.1 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 21
!
interface Serial1/0.2 point-to-point
 ip vrf forwarding vrf2
 ip address 10.3.2.1 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 22
!
interface Serial1/0.151 point-to-point
```

```
 ip address 10.7.151.1 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 151
!
interface Serial1/0.152 point-to-point
 ip address 10.7.152.1 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 152
!
interface Serial1/1
 no ip address
 no ip mroute-cache
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 encapsulation frame-relay
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 no keepalive
 serial restart-delay 0
 no frame-relay inverse-arp
!
interface Serial1/2.1 point-to-point
 ip vrf forwarding vrf1
 ip address 10.1.1.2 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 21
 crypto map vrf1-1
!
interface Serial1/2.2 point-to-point
 ip vrf forwarding vrf2
 ip address 10.1.2.2 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 22
 crypto map vrf2-1
!
interface Serial1/2.151 point-to-point
 ip address 10.5.151.2 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 151
 crypto map global1-1
!
interface Serial1/2.152 point-to-point
 ip address 10.5.152.2 255.255.255.0
 no ip route-cache
 frame-relay interface-dlci 152
 crypto map global2-1
!
interface Serial1/3
 no ip address
 no ip mroute-cache
 shutdown
 serial restart-delay 0
!
ip default-gateway 10.1.38.1
ip classless
ip route 10.1.1.6 255.255.255.255 10.1.151.1
ip route 10.2.1.6 255.255.255.255 10.1.152.1
ip route 10.6.2.1 255.255.255.255 10.7.151.2
ip route 10.6.2.2 255.255.255.255 10.7.152.2
ip route 172.19.216.110 255.255.255.255 FastEthernet0/0
```

```
ip route vrf vrf1 10.20.1.1 255.255.255.255 10.1.1.1
ip route vrf vrf1 10.22.1.1 255.255.255.255 10.30.1.1
ip route vrf vrf2 10.20.2.1 255.255.255.255 10.1.2.1
ip route vrf vrf2 10.22.2.1 255.255.255.255 10.30.1.2
!
!
ip http server
no ip http secure-server
!
ip access-list standard vrf-vrf1-context
ip access-list standard vrf-vrf2-context
!
access-list 101 permit ip host 10.22.1.1 host 10.20.1.1
access-list 102 permit ip host 10.22.2.1 host 10.20.2.1
access-list 151 permit ip host 10.6.2.1 host 10.1.1.6
access-list 152 permit ip host 10.6.2.2 host 10.2.1.6
snmp-server group abc1 v2c context vrf-vrf1-context read view_vrf1 notify
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F access vrf-vrf1-context
snmp-server group abc2 v2c context vrf-vrf2-context read view_vrf2 notify
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F access vrf-vrf2-context
snmp-server view view_vrf1 iso included
snmp-server view view_vrf2 iso included
snmp-server community abc1 RW
snmp-server community global1 RW
snmp-server community abc2 RW
snmp-server community global2 RW
snmp-server enable traps tty
snmp-server enable traps config
snmp-server host 172.19.216.110 version 2c abc1
snmp-server host 172.19.216.110 vrf vrf1 version 2c abc1 udp-port 2001  ipsec isakmp
snmp-server host 172.19.216.110 version 2c abc2
snmp-server host 172.19.216.110 vrf vrf2 version 2c abc2 udp-port 2002  ipsec isakmp
snmp-server context vrf-vrf1-context
snmp-server context vrf-vrf2-context
!
!
snmp mib community-map  abc1 context vrf-vrf1-context
snmp mib community-map  abc2 context vrf-vrf2-context
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
!
webvpn context Default_context
 ssl authenticate verify all
 !
 no inservice
!
!
end
```

Both VRFs Cleared

The following output, for abc1 and abc2, shows that both VRFs have been “cleared” to ensure that all the counters are initialized to a known value.

The following output shows that VRF abc1 has been cleared:

```
orcas:2> setenv SR_MGR_CONF /users/green1
orcas:3> setenv SR_UTIL_SNMP_VERSION v2c
orcas:5> setenv SR_UTIL_COMMUNITY abc1
orcas:6> setenv SR_MGR_CONF_DIR /users/green1
orcas:7> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
```

```
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
```

The following output shows that VRF abc2 has been cleared:

```
orcas:8> setenv SR_UTIL_COMMUNITY abc2
orcas:9> /auto/sw/packages/snmpr/14.2.0.0/solaris2bin/getmany -v2c 10.1.38.25 cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
```

```
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:10>
orcas:10>
orcas:10>
```

VRF abc1 Pinged

The following output shows that VRF abc1 has been pinged:

```
Router3745a# ping
Protocol [ip]:
Target IP address: 10.22.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.20.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.22.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.20.1.1
```

VRF abc1 Polled

Polling VRF abc1 results in the following output:

Note After the ping, the counters should show some nonzero values.

```
orcas:10>
orcas:12> setenv SR_UTIL_COMMUNITY abc1
orcas:13> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 1
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 336
cikeGlobalInPkts.0 = 2
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 1
cikeGlobalInP2Exchgs.0 = 2
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 344
cikeGlobalOutPkts.0 = 2
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 1
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cikePeerLocalAddr.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.49.1
 = 0a 01  01 02
cikePeerRemoteAddr.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.49.1
 = 0a 01  01 01
cikePeerActiveTime.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.49.1
 = 13743
cikePeerActiveTunnelIndex.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.49.1
 = 1
cikeTunLocalType.1 = ipAddrPeer(1)
cikeTunLocalValue.1 = 010.001.001.002
cikeTunLocalAddr.1 = 0a 01  01 02
cikeTunLocalName.1 = ipsecf-3745b
cikeTunRemoteType.1 = ipAddrPeer(1)
cikeTunRemoteValue.1 = 010.001.001.001
cikeTunRemoteAddr.1 = 0a 01  01 01
cikeTunRemoteName.1 =
cikeTunNegoMode.1 = main(1)
cikeTunDiffHellmanGrp.1 = dhGroup1(2)
cikeTunEncryptAlgo.1 = des(2)
cikeTunHashAlgo.1 = sha(3)
cikeTunAuthMethod.1 = preSharedKey(2)
cikeTunLifeTime.1 = 86400
cikeTunActiveTime.1 = 13752
cikeTunSaRefreshThreshold.1 = 0
cikeTunTotalRefreshes.1 = 0
cikeTunInOctets.1 = 336
cikeTunInPkts.1 = 2
```

```
cikeTunInDropPkts.1 = 0
cikeTunInNotifys.1 = 1
cikeTunInP2Exchgs.1 = 2
cikeTunInP2ExchgInvalids.1 = 0
cikeTunInP2ExchgRejects.1 = 0
cikeTunInP2SaDelRequests.1 = 0
cikeTunOutOctets.1 = 344
cikeTunOutPkts.1 = 2
cikeTunOutDropPkts.1 = 0
cikeTunOutNotifys.1 = 0
cikeTunOutP2Exchgs.1 = 1
cikeTunOutP2ExchgInvalids.1 = 0
cikeTunOutP2ExchgRejects.1 = 0
cikeTunOutP2SaDelRequests.1 = 0
cikeTunStatus.1 = active(1)
cikePeerCorrIpSecTunIndex.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.49.1.1
 = 1
cipSecGlobalActiveTunnels.0 = 1
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 400
cipSecGlobalHcInOctets.0 = 0x0190
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 400
cipSecGlobalHcInDecompOctets.0 = 0x0190
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 4
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 4
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 4
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 704
cipSecGlobalHcOutOctets.0 = 0x02c0
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 704
cipSecGlobalHcOutUncompOctets.0 = 0x02c0
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 4
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 4
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 4
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecTunIkeTunnelIndex.1 = 1
cipSecTunIkeTunnelAlive.1 = true(1)
cipSecTunLocalAddr.1 = 0a 01  01 02
cipSecTunRemoteAddr.1 = 0a 01  01 01
cipSecTunKeyType.1 = ike(1)
cipSecTunEncapMode.1 = tunnel(1)
cipSecTunLifeSize.1 = 99000
cipSecTunLifeTime.1 = 5000
cipSecTunActiveTime.1 = 13749
cipSecTunSaLifeSizeThreshold.1 = 64
cipSecTunSaLifeTimeThreshold.1 = 10
cipSecTunTotalRefreshes.1 = 0
cipSecTunExpiredSaInstances.1 = 0
cipSecTunCurrentSaInstances.1 = 4
cipSecTunInSaDiffHellmanGrp.1 = dhGroup1(2)
cipSecTunInSaEncryptAlgo.1 = des(2)
cipSecTunInSaAhAuthAlgo.1 = hmacSha(3)
```

```
cipSecTunInSaEspAuthAlgo.1 = hmacSha(3)
cipSecTunInSaDecompAlgo.1 = none(1)
cipSecTunOutSaDiffHellmanGrp.1 = dhGroup1(2)
cipSecTunOutSaEncryptAlgo.1 = des(2)
cipSecTunOutSaAhAuthAlgo.1 = hmacSha(3)
cipSecTunOutSaEspAuthAlgo.1 = hmacSha(3)
cipSecTunOutSaCompAlgo.1 = none(1)
cipSecTunInOctets.1 = 400
cipSecTunHcInOctets.1 = 0x0190
cipSecTunInOctWraps.1 = 0
cipSecTunInDecompOctets.1 = 400
cipSecTunHcInDecompOctets.1 = 0x0190
cipSecTunInDecompOctWraps.1 = 0
cipSecTunInPkts.1 = 4
cipSecTunInDropPkts.1 = 0
cipSecTunInReplayDropPkts.1 = 0
cipSecTunInAuths.1 = 4
cipSecTunInAuthFails.1 = 0
cipSecTunInDecrypts.1 = 4
cipSecTunInDecryptFails.1 = 0
cipSecTunOutOctets.1 = 704
cipSecTunHcOutOctets.1 = 0x02c0
cipSecTunOutOctWraps.1 = 0
cipSecTunOutUncompOctets.1 = 704
cipSecTunHcOutUncompOctets.1 = 0x02c0
cipSecTunOutUncompOctWraps.1 = 0
cipSecTunOutPkts.1 = 4
cipSecTunOutDropPkts.1 = 0
cipSecTunOutAuths.1 = 4
cipSecTunOutAuthFails.1 = 0
cipSecTunOutEncrypts.1 = 4
cipSecTunOutEncryptFails.1 = 0
cipSecTunStatus.1 = active(1)
cipSecEndPtLocalName.1.1 =
cipSecEndPtLocalType.1.1 = singleIpAddr(1)
cipSecEndPtLocalAddr1.1.1 = 16 01  01 01
cipSecEndPtLocalAddr2.1.1 = 16 01  01 01
cipSecEndPtLocalProtocol.1.1 = 0
cipSecEndPtLocalPort.1.1 = 0
cipSecEndPtRemoteName.1.1 =
cipSecEndPtRemoteType.1.1 = singleIpAddr(1)
cipSecEndPtRemoteAddr1.1.1 = 14 01  01 01
cipSecEndPtRemoteAddr2.1.1 = 14 01  01 01
cipSecEndPtRemoteProtocol.1.1 = 0
cipSecEndPtRemotePort.1.1 = 0
cipSecSpiDirection.1.1 = in(1)
cipSecSpiDirection.1.2 = out(2)
cipSecSpiDirection.1.3 = in(1)
cipSecSpiDirection.1.4 = out(2)
cipSecSpiValue.1.1 = 3891970674
cipSecSpiValue.1.2 = 1963217493
cipSecSpiValue.1.3 = 3691920464
cipSecSpiValue.1.4 = 3458912974
cipSecSpiProtocol.1.1 = ah(1)
cipSecSpiProtocol.1.2 = ah(1)
cipSecSpiProtocol.1.3 = esp(2)
cipSecSpiProtocol.1.4 = esp(2)
cipSecSpiStatus.1.1 = active(1)
cipSecSpiStatus.1.2 = active(1)
cipSecSpiStatus.1.3 = active(1)
cipSecSpiStatus.1.4 = active(1)
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
```

```
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:14>
orcas:14>
orcas:14>
```

VRF abc2 Polled

Polling VRF abc2 results in the following output:

Note The ping was completed for VRF abc1 only. Therefore, the counters of VRF abc2 should remain in the initialized state.

```
setenv SR_UTIL_COMMUNITY abc2
orcas:15>
orcas:15> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
```

```
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:16>
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands by technology	Cisco IOS Release Command References
Cisco IOS master commands list	Master Command List
Configuring SNMP	The chapter “Configuring SNMP Support”in the *Cisco IOS Network Management Configuration Guide.*
Configuring VRF-Aware IPsec	VRF-Aware IPSec

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR-MIB • CISCO-IPSEC-MIB • The CISCO-IPSEC-POLICY-MAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec and IKE MIB Support for Cisco VRF-Aware IPsec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 252: Feature Information for Feature Information for IPsec and IKE MIB Support for Cisco VRF-Aware IPsec

Feature Name	Releases	Feature Information
IPsec and IKE MIB Support for Cisco VRF-Aware IPsec	IOS XE 3.1S	The IPsec and IKE MIB Support for the Virtual Private Network routing and forwarding- (VRF-) aware IP security (IPsec) feature allows VRF-aware IPsec to be managed with MIBs, which provide the details of IPsec statistics and performance metrics on a per VRF basis. This feature was introduced in Cisco IOS Release 12.4(4)T. This feature was integrated into Cisco IOS Release XE 3.1S. The following commands were introduced or modified: **debug crypto mib**.

CHAPTER 189

IPsec SNMP Support

The IP Security (IPsec) SNMP Support feature introduces support for industry-standard IPsec MIBs and Cisco IOS XE-software specific IPsec MIBs.

The commands in this feature allow you to examine the version of the IPsec MIB feature, to enable and disable SNMP traps, and to monitor and control the size of the buffers used by this feature.

Note This document focuses on Cisco IOS XE CLI support for the Cisco IPsec MIBs. This document also lists which elements of the MIBs are currently supported. This document does not describe SNMP configuration (from a Network Management Station) of the Cisco IPsec MIBs.

Restrictions for IPsec SNMP Support

- Only the following tunnel setup failure logs are supported with the IPsec--SNMP Support feature:
  - NOTIFY_MIB_IPSEC_PROPOSAL_INVALID
  - “A tunnel could not be established because the peer did not supply an acceptable proposal.”
  - NOTIFY_MIB_IPSEC_ENCRYPT_FAILURE
  - “A tunnel could not be established because it failed to encrypt a packet to be sent to a peer.”
  - NOTIFY_MIB_IPSEC_SYSCAP_FAILURE
  - “A tunnel could not be established because the system ran out of resources.”
  - NOTIFY_MIB_IPSEC_LOCAL_FAILURE
  - “A tunnel could not be established because of an internal error.”

Note that these failure notices are recorded in the failure tables, but are not available as SNMP notifications (traps).

- The following functions are not supported with the IPsec MIB feature:
  - Checkpointing
  - The Dynamic Cryptomap table of the CISCO-IPSEC-MIB
- The CISCO-IPSEC-POLICY-MAP-MIB (ciscoIpSecPolMap) defines no notifications (the "IPSec Policy Map Notifications Group" is empty).

Information About IPsec SNMP Support

The IP Security (IPsec) SNMP Support feature introduces support for industry-standard IPsec MIBs and Cisco IOS XE-software specific IPsec MIBs.

The IPsec MIBs allow IPsec configuration monitoring and IPsec status monitoring using SNMP, and can be integrated in a variety of Virtual Private Network (VPN) management solutions.

For example, this feature allows you to specify the desired size of a tunnel history table or a tunnel failure table using the Cisco IOS XE CLI. The history table archives attribute and statistic information about the tunnel; the failure table archives tunnel failure reasons along with the time failure occurred. A failure history table can be used as a simple method to distinguish between a normal and an abnormal tunnel termination. That is, if a tunnel entry in the tunnel history table has no associated failure record, the tunnel must have terminated normally. However, a tunnel history table does not accompany every failure table because every failure does not correspond to a tunnel. Thus, supported setup failures are recorded in the failure table, but an associated history table is not recorded because a tunnel was never set up.

This feature also provides IPsec Simple Network Management Protocol (SNMP) notifications for use with network management systems.

Related Features and Technologies

The IPsec--SNMP Support feature was designed to support the VPN Device Manager (VDM). VDM enables network administrators to manage and configure site-to-site VPNs on a single device from a web browser and to see the effects of changes in real time. VDM implements a wizard-based graphical user interface (GUI) to simplify the process of configuring site-to-site VPNs using the IPsec protocol. VDM software is installed directly on Cisco VPN routers, and is designed for use and compatibility with future Device Manager products.

How to Configure IPsec SNMP Support

Enabling IPsec SNMP Notifications

To enable IPsec SNMP notifications, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps ipsec cryptomap** [**add** | **delete** | **attach** | **detach**]
4. **snmp-server enable traps isakmp** [**policy** {**add** | **delete**} | **tunnel** {**start** | **stop**}]

5. **snmp-server host** *host-address* **traps** *community-string* **ipsec**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps ipsec cryptomap** [**add** | **delete** | **attach** | **detach**] **Example:** `Router (config)# snmp-server enable traps ipsec cryptomap add`	Enables a router to send IPsec SNMP notifications.
Step 4	**snmp-server enable traps isakmp** [**policy** {**add** | **delete**} | **tunnel** {**start** | **stop**}] **Example:** `Router (config)# snmp-server enable traps isakmp policy add`	Enables a router to send IPsec ISAKMP SNMP notifications.
Step 5	**snmp-server host** *host-address* **traps** *community-string* **ipsec** **Example:** `Router (config)# snmp-server host my.example.com traps version2c`	Specifies the recipient of IPsec SNMP notification operations.

What to do next

For more information on configuring SNMP, refer to the chapter "Configuring SNMP Support" in the *Cisco IOS XE Configuration Fundamentals Configuration Guide* .

Configuring IPsec Failure History Table Size

The default failure history table size is 200. To change the size of the failure history table, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **crypto mib ipsec flowmib history failure size** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto mib ipsec flowmib history failure size** *number* **Example:** `Router (config)# crypto mib ipsec flowmib history failure size 220`	Changes the size of the IPsec failure history table.

Configuring IPsec Tunnel History Table Size

The default tunnel history table size is 200. To change the size of the tunnel history table, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto mib ipsec flowmib history tunnel size** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto mib ipsec flowmib history tunnel size** *number* **Example:** `Router (config)# crypto mib ipsec flowmib history tunnel size`	Changes the size of the IPsec tunnel history table.

Verifying IPsec MIB Configuration

To verify that the IPsec MIB feature is configured properly, perform the following tasks:

- Enter the **show crypto mib ipsec flowmib history failure size**privileged EXEC commandto display the size of the failure history table:

```
Router# show crypto mib ipsec flowmib history failure size
IPSec Failure Window Size: 140
```

- Enter the **show crypto mib ipsec flowmib history tunnel size** privileged EXEC command to display the size of the tunnel history table**:**

```
Router# show crypto mib ipsec flowmib history tunnel size
IPSec History Window Size: 130
```

- Enter the **show crypto mib ipsec flowmib version**privileged EXEC command to display the MIB version used by the management applications to identify the feature set:

```
Router# show crypto mib ipsec flowmib version
IPSec Flow MIB version: 1
```

- Enter the **debug crypto mib** command to display the IPsec MIB debug message notifications:

```
Router# debug crypto mib
Crypto IPSec Mgmt Entity debugging is on
```

Monitoring and Maintaining IPsec MIB

To monitor the status of IPsec MIB information, use any of the following commands.

SUMMARY STEPS

1. **enable**
2. **show crypto mib ipsec flowmib history failure size**
3. **show crypto mib ipsec flowmib history tunnel size**
4. **show crypto mib ipsec flowmib version**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto mib ipsec flowmib history failure size** **Example:** `Router# show crypto mib ipsec flowmib history failure size`	Displays the size of the IPsec failure history table.
Step 3	**show crypto mib ipsec flowmib history tunnel size** **Example:** `Router# show crypto mib ipsec flowmib history tunnel size`	Displays the size of the IPsec tunnel history table.
Step 4	**show crypto mib ipsec flowmib version** **Example:** `Router# show crypto mib ipsec flowmib version`	Displays the IPsec Flow MIB version used by the router.

Configuration Examples for IPsec SNMP Support

Enabling IPsec Notifications Examples

In the following example, IPsec notifications are enabled:

```
snmp-server enable traps ipsec isakmp
```

In the following example, the router is configured to send IPsec notifications to the host nms1.example.com:

```
snmp-server host nms1.example.com public ipsec isakmp
Translating "nms1.example.com"...domain server (172.00.0.01) [OK]
```

Specifying History Table Size Examples

In the following example, the specified failure history table size is 140:

```
crypto mib ipsec flowmib history failure size 140
```

In the following example, the specified tunnel history table size is 130:

```
crypto mib ipsec flowmib history tunnel size 130
```

Additional References

Related Documents

Related Topic	Document Title
Configuring AAA accounting	• Configuring Accounting
Configuring IPsec VPN accounting	• Configuring Security for VPNs with IPsec
Configuring basic AAA RADIUS	• The section "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide: User Services* on Cisco.com
Configuring ISAKMP profiles	VRF Aware IPsec
Privilege levels with TACACS+ and RADIUS	• Configuring TACACS+ • "Configuring RADIUS" section of the *Cisco IOS Security Configuration Guide: User Services on Cisco.com*
IP security, RADIUS, and AAA commands	*Cisco IOS Security Command Reference*
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec SNMP Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 253: Feature Information for IPsec SNMP Support

Feature Name	Releases	Feature Information
IPsec SNMP Support	Cisco IOS XE Release 2.1	The IP Security (IPsec) SNMP Support feature introduces support for industry-standard IPsec MIBs and Cisco IOS XE-software specific IPsec MIBs. The following commands were introduced or modified: **crypto mib ipsec flowmib history failure size, crypto mib ipsec flowmib history tunnel size, debug crypto mib, show crypto mib ipsec flowmib history failure size, show crypto mib ipsec flowmib history tunnel size, show crypto mib ipsec flowmib version, snmp-server enable traps ipsec, snmp-server enable traps isakmp, snmp-server host**.

Glossary

CA --certificate authority. A certificate authority (CA) is an entity in a network that issues and manages security credentials and public keys (in the form of X509v3 certificates) for message encryption. As part of a public key infrastructure (PKI), a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. If the RA verifies the requestor's information, the CA can then issue a certificate. Certificates generally include the owner’s public key, the expiration date of the certificate, the owner’s name, and other information about the public key owner.

IP Security --See IPsec.

IPsec --Internet Protocol Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Management Information Base --See MIB.

MIB --Management Information Base. Database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP) or Common Management Information Protocol (MIP). The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a graphical user interface (GUI) network management system (NMS). MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

Simple Network Management Protocol --See SNMP.

SNMP --Simple Network Management Protocol. An application-layer protocol that provides a message format for communication between SNMP managers and agents.

trap --Message sent by an SNMP agent to a network management system, console, or terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

CHAPTER 190

IPsec VPN Accounting

The IPsec VPN Accounting feature allows for a session to be accounted for by indicating when the session starts and when it stops.

A VPN session is defined as an Internet Key Exchange (IKE) security association (SA) and the one or more SA pairs that are created by the IKE SA. The session starts when the first IP Security (IPsec) pair is created and stops when all IPsec SAs are deleted.

Session identifying information and session usage information is passed to the Remote Authentication Dial-In User Service (RADIUS) server through standard RADIUS attributes and vendor-specific attributes (VSAs).

Prerequisites for IPsec VPN Accounting

- Understand how to configure RADIUS and authentication, authorization, and accounting (AAA) accounting.
- Understand how to configure IPsec accounting.

Information About IPsec VPN Accounting

RADIUS Accounting

For many large networks, it is required that user activity be recorded for auditing purposes. The method that is used most is RADIUS accounting.

RADIUS accounting allows for a session to be accounted for by indicating when the session starts and when it stops. Additionally, session identifying information and session usage information is passed to the RADIUS server through RADIUS attributes and VSAs.

RADIUS Start Accounting

The RADIUS Start packet contains many attributes that generally identify who is requesting the service and of what the property of that service consists. The table below represents the attributes required for the start.

Table 254: RADIUS Accounting Start Packet Attributes

RADIUS Attributes Value	Attribute	Description
1	user-name	Username used in extended authentication (XAUTH).The username may be NULL when XAUTH is not used.
4	nas-ip-address	Identifying IP address of the network access server (NAS) that serves the user. It should be unique to the NAS within the scope of the RADIUS server.
5	nas-port	Physical port number of the NAS that serves the user.
8	framed-ip-address	Private address allocated for the IP Security (IPsec) session.
40	acct-status-type	Status type. This attribute indicates whether this accounting request marks the beginning (start), the end (stop), or an update of the session.
41	acct-delay-time	Number of seconds the client has been trying to send a particular record.
44	acct-session-id	Unique accounting identifier that makes it easy to match start and stop records in a log file.
26	vrf-id	String that represents the name of the Virtual Route Forwarder (VRF).
26	isakmp-initiator-ip	Endpoint IP address of the remote Internet Key Exchange (IKE) initiator (V4).
26	isakmp-group-id	Name of the VPN group profile used for accounting.
26	isakmp-phase1-id	Phase 1 identification (ID) used by IKE (for example, domain name [DN], fully qualified domain name [FQDN], IP address) to help identify the session initiator.

RADIUS Stop Accounting

The RADIUS Stop packet contains many attributes that identify the usage of the session. Table 2 represents the additional attributes required for the RADIUS stop packet. It is possible that only the stop packet is sent without the start if configured to do so. If only the stop packet is sent, this allows an easy way to reduce the number of records going to the AAA server.

Table 255: RADIUS Accounting Stop Packet Attributes

RADIUS Attributes Value	Attribute	Description
42	acct-input-octets	Number of octets that have been received from the Unity client over the course of the service that is being provided.
43	acct-output-octets	Number of octets that have been sent to the Unity client in the course of delivering this service.
46	acct-session-time	Length of time (in seconds) that the Unity client has received service.
47	acct-input-packets	Quantity of packets that have been received from the Unity client in the course of delivering this service.
48	acct-output-packets	Quantity of packets that have been sent to the Unity client in the course of delivering this service.
49	acct-terminate-cause	For future use.
52	acct-input-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 232 (2 to the 32nd power) over the course of this service.
52	acct-output-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 232 (2 to the 32nd power) over the course of this service.

RADIUS Update Accounting

RADIUS accounting updates are supported. Packet and octet counts are shown in the updates.

IKE and IPsec Subsystem Interaction

Accounting Start

If IPsec accounting is configured, after IKE phases are complete, an accounting start record is generated for the session. New accounting records are not generated during a rekeying.

The following is an account start record that was generated on a router and that is to be sent to the AAA server that is defined:

```
*Aug 23 04:06:20.131: RADIUS(00000002): sending
*Aug 23 04:06:20.131: RADIUS(00000002): Send Accounting-Request to 10.1.1.4:1646 id 4, len
 220
*Aug 23 04:06:20.131: RADIUS:  authenticator 38 F5 EB 46 4D BE 4A 6F - 45 EB EF 7D B7 19
FB 3F
*Aug 23 04:06:20.135: RADIUS:  Acct-Session-Id     [44]  10  "00000001"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  31
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   25  "isakmp-group-id=cclient"
```

```
*Aug 23 04:06:20.135: RADIUS:  Framed-IP-Address   [8]   6   10.13.13.1
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=10.1.2.2"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No Progress"
*Aug 23 04:06:20.135: RADIUS:  User-Name           [1]   13  "username1"
*Aug 23 04:06:20.135: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:06:20.135: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 04:06:20.135: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:06:20.135: RADIUS:  NAS-IP-Address      [4]   6   10.1.1.147
*Aug 23 04:06:20.135: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:06:20.139: RADIUS: Received from id 21645/4 10.1.1.4:1646, Accounting-response,
 len 20
*Aug 23 04:06:20.139: RADIUS:  authenticator B7 E3 D0 F5 61 9A 89 D8 - 99 A6 8A 8A 98 79
9D 5D
```

Accounting Stop

An accounting stop packet is generated when there are no more flows (IPsec SA pairs) with the remote peer.

The accounting stop records contain the following information:

- Packets out
- Packets in
- Octets out
- Gigawords in
- Gigawords out

Below is an account start record that was generated on a router. The account start record is to be sent to the AAA server that is defined.

```
*Aug 23 04:20:16.519: RADIUS(00000003): Using existing nas_port 0
*Aug 23 04:20:16.519: RADIUS(00000003): Config NAS IP: 100.1.1.147
*Aug 23 04:20:16.519: RADIUS(00000003): sending
*Aug 23 04:20:16.519: RADIUS(00000003): Send Accounting-Request to 100.1.1.4:1646 id 19,
len 238
*Aug 23 04:20:16.519: RADIUS:  authenticator 82 65 5B 42 F0 3F 17 C3 - 23 F3 4C 35 A2 8A
3E E6
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=10.1.1.2"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No Progress"
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Time   [46]  6   709
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Octets   [42]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Octets  [43]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Packets  [47]  6   1004
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Packets [48]  6   1004
*Apr 23 04:20:16.519: RADIUS:  Acct-Input-Giga-Word[52]  6   0
*Apr 23 04:20:16.519: RADIUS:  Acct-Output-Giga-Wor[53]  6   0

*Aug 23 04:20:16.519: RADIUS:  Acct-Terminate-Cause[49]  6   none                      [0]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  32
```

```
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   26  "disc-cause-ext=No Reason"
*Aug 23 04:20:16.519: RADIUS:  Acct-Status-Type    [40]  6   Stop                      [2]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:20:16.519: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 04:20:16.519: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:20:16.519: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 04:20:16.519: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:20:16.523: RADIUS: Received from id 21645/19 100.1.1.4:1646, Accounting-response,
 len 20
*Aug 23 04:20:16.523: RADIUS:  authenticator F1 CA C1 28 CE A0 26 C9 - 3E 22 C9 DA EA B8
22 A0
```

Accounting Updates

If accounting updates are enabled, accounting updates are sent while a session is “up.” The update interval is configurable. To enable the accounting updates, use the **aaa accounting update** command.

The following is an accounting update record that is being sent from the router:

```
Router#
*Aug 23 21:46:05.263: RADIUS(00000004): Using existing nas_port 0
*Aug 23 21:46:05.263: RADIUS(00000004): Config NAS IP: 100.1.1.147
*Aug 23 21:46:05.263: RADIUS(00000004): sending
*Aug 23 21:46:05.263: RADIUS(00000004): Send Accounting-Request to 100.1.1.4:1646 id 22,
len 200
*Aug 23 21:46:05.263: RADIUS:  authenticator 30 FA 48 86 8E 43 8E 4B - F9 09 71 04 4A F1
52 25
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Id     [44]  10  "00000003"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=10.1.1.2"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair      [1]   30  "connect-progress=No Progress"
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Time   [46]  6   109
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Octets   [42]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Octets  [43]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Packets  [47]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Packets [48]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Status-Type    [40]  6   Watchdog                  [3]
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 21:46:05.263: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 21:46:05.263: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 21:46:05.263: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 21:46:05.263: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 21:46:05.267: RADIUS: Received from id 21645/22 100.1.1.4:1646, Accounting-response,
 len 20
*Aug 23 21:46:05.267: RADIUS:  authenticator 51 6B BB 27 A4 F5 D7 61 - A7 03 73 D3 0A AC
1C
```

How to Configure IPsec VPN Accounting

Configuring IPsec VPN Accounting

Before you begin

IPsec must be configured first before configuring IPsec VPN accounting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** *list-name method*
5. **aaa authorization network** *list-name method*
6. **aaa accounting network list-name start-stop** [**broadcast**] **group** *group-name*
7. **aaa session-id common**
8. **crypto isakmp profile** *profile-name*
9. **vrf** *ivrf*
10. **match identity group** *group-name*
11. **client authentication list** *list-name*
12. **isakmp authorization list** *list-name*
13. **client configuration address** [**initiate** | **respond**]
14. **accounting** *list-name*
15. **exit**
16. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
17. **set transform-set** *transform-set-name*
18. **set isakmp-profile** *profile-name*
19. **reverse-route** [**remote-peer**]
20. **exit**
21. **crypto map** *map-name* **ipsec-isakmp dynamic** *dynamic-template-name*
22. **radius-server host** *ip-address* [**auth-port** *port-number]* [**acct-port** *port-number*]
23. **radius-server key** *string*
24. **radius-server vsa send accounting**
25. **interface** *type slot / port*
26. **crypto map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables periodic interim accounting records to be sent to the accounting server.

	Command or Action	Purpose
Step 4	**aaa authentication login** *list-name method* **Example:** `Router (config)# aaa authentication login cisco-client group radius`	Enforces authentication, authorization, and accounting (AAA) authentication for extended authorization (XAUTH) through RADIUS or local.
Step 5	**aaa authorization network** *list-name method* **Example:** `Router (config)# aaa authorization network cisco-client group radius`	Sets AAA authorization parameters on the remote client from RADIUS or local.
Step 6	**aaa accounting network list-name start-stop** [**broadcast**] **group** *group-name* **Example:** `Router (config)# aaa accounting network acc start-stop broadcast group radius`	Enables AAA accounting of requested services for billing or security purposes when RADIUS or TACACS+ is used.
Step 7	**aaa session-id common** **Example:** `Router (config)# aaa session-id common`	Specifies whether the same session ID is used for each AAA accounting service type within a call or whether a different session ID is assigned to each accounting service type.
Step 8	**crypto isakmp profile** *profile-name* **Example:** `Route (config)# crypto isakmp profile cisco`	Audits IP security (IPsec) user sessions and enters isakmp-profile submode.
Step 9	**vrf** *ivrf* **Example:** `Router (conf-isa-prof)# vrf cisco`	Associates the on-demand address pool with a Virtual Private Network (VPN) routing and forwarding (VRF) instance name.
Step 10	**match identity group** *group-name* **Example:** `Router(conf-isa-prof)# match identity group cisco`	Matches an identity from a peer in an ISAKMP profile.
Step 11	**client authentication list** *list-name* **Example:** `Router(conf-isa-prof)# client authentication list cisco`	Configures Internet Key Exchange (IKE) extended authentication (XAUTH) in an Internet Security Association and Key Management Protocol (ISAKMP) profile.
Step 12	**isakmp authorization list** *list-name* **Example:**	Configures an IKE shared secret and other parameters using the AAA server in an ISAKMP profile. The shared

	Command or Action	Purpose
	`Router(conf-isa-prof)# isakmp authorization list cisco-client`	secret and other parameters are generally pushed to the remote peer through mode configuration (MODECFG).
Step 13	**client configuration address** [**initiate** | **respond**] **Example:** `Router(conf-isa-prof)# client configuration address respond`	Configures IKE mode configuration (MODECFG) in the ISAKMP profile.
Step 14	**accounting** *list-name* **Example:** `Router(conf-isa-prof)# accounting acc`	Enables AAA accounting services for all peers that connect through this ISAKMP profile.
Step 15	**exit** **Example:** `Router(conf-isa-prof)# exit`	Exits isakmp-profile submode.
Step 16	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map mymap 10 ipsec-isakmp`	Creates a dynamic crypto map template and enters the crypto map configuration command mode.
Step 17	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set aswan`	Specifies which transform sets can be used with the crypto map template.
Step 18	**set isakmp-profile** *profile-name* **Example:** `Router(config-crypto-map)# set isakmp-profile cisco`	Sets the ISAKMP profile name.
Step 19	**reverse-route** [**remote-peer**] **Example:** `Router(config-crypto-map)# reverse-route`	Allows routes (ip addresses) to be injected for destinations behind the VPN remote tunnel endpoint and may include a route to the tunnel endpoint itself (using the **remote-peer** keyword for the crypto map.
Step 20	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits dynamic crypto map configuration mode.

	Command or Action	Purpose
Step 21	**crypto map** *map-name* **ipsec-isakmp dynamic** *dynamic-template-name* **Example:** `Router(config)# crypto map mymap ipsec-isakmp dynamic dmap`	Enters crypto map configuration mode
Step 22	**radius-server host** *ip-address* [**auth-port** *port-number]* [**acct-port** *port-number*] **Example:** `Router(config)# radius-server host 172.16.1.4`	Specifies a RADIUS server host.
Step 23	**radius-server key** *string* **Example:** `Router(config)# radius-server key nsite`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 24	**radius-server vsa send accounting** **Example:** `Router(config)# radius-server vsa send accounting`	Configures the network access server to recognize and use vendor-specific attributes.
Step 25	**interface** *type slot / port* **Example:** `Router(config)# interface FastEthernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 26	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a previously defined crypto map set to an interface.

Configuring Accounting Updates

To send accounting updates while a session is "up," perform the following optional task:

Before you begin

IPsec VPN accounting must be configured before accounting updates are configured. See Configuring IPsec VPN Accounting, on page 2459 for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting update periodic** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting update periodic** *number* **Example:** `Router (config)# aaa accounting update periodic 1-2147483647`	(Optional) Enables periodic interim accounting records to be sent to the accounting server.

Troubleshooting for IPsec VPN Accounting

To display messages about IPsec accounting events, perform the following optional task:

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp aaa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp aaa** **Example:** `Router# debug crypto isakmp aaa`	Displays messages about Internet Key Exchange (IKE) events. • The **aaa** keyword specifies accounting events.

Configuration Examples for IPsec VPN Accounting

Accounting and ISAKMP-Profile Example

The following example shows a configuration for supporting remote access clients with accounting and ISAKMP profiles:

```
version 2.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
authentication pre-share
group 2
!
crypto isakmp policy 10
hash md5
authentication pre-share
lifetime 200
crypto isakmp key cisco address 172.31.100.2
crypto iakmp client configuration group cclient
 key jegjegjhrg
 pool addressA

crypto-isakmp profile groupA
 vrf cisco
 match identity group cclient
 client authentication list cisco-client
 isakmp authorization list cisco-client
 client configuration address respond
 accounting acc
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto dynamic-map remotes 1
set peer 172.31.100.2
set security-association lifetime seconds 120
set transform-set esp-des-md5
reverse-route
```

```
!
crypto map test 10 ipsec-isakmp dynamic remotes
!
voice call carrier capacity active
!
interface Loopback0
ip address 10.20.20.20 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface FastEthernet0/0
ip address 10.2.80.203 255.255.255.0
no ip mroute-cache
load-interval 30
duplex full
!
interface FastEthernet1/0
ip address 192.168.219.2 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
!
interface FastEthernet1/1
ip address 172.28.100.1 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.20.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73
ip local pool addressA 192.168.1.1 192.168.1.253
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
shutdown
!
!
line con 0
```

```
exec-timeout 0 0
exec prompt timestamp
line aux 0
line vty 5 15
 ntp server 172.31.150.52
end
```

Accounting Without ISAKMP Profiles Example

The following example shows a full Cisco IOS XE configuration that supports accounting remote access peers when ISAKMP profiles are not used:

```
version 2.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 lifetime 200
crypto isakmp key cisco address 172.31.100.2
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto map test client accounting list ipsecaaa
crypto map test 10 ipsec-isakmp
 set peer 172.31.100.2
 set security-association lifetime seconds 120
 set transform-set esp-des-md5
 match address 101
!
voice call carrier capacity active
!
interface Loopback0
 ip address 10.20.20.20 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
```

```
interface FastEthernet0/0
 ip address 10.2.80.203 255.255.255.0
 no ip mroute-cache
 load-interval 30
 duplex full
!
interface FastEthernet1/0
 ip address 192.168.219.2 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 172.28.100.1 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.30.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
 permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 exec prompt timestamp
line aux 0
line vty 5 15
!
exception core-file ioscrypto/core/sheep-core
exception dump 172.25.1.129
ntp clock-period 17208229
ntp server 172.71.150.52
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Configuring AAA accounting	" Configuring Accounting " module in the *Cisco IOS XE Security Configuration Guide: Securing User Services*
Configuring IPsec VPN accounting	" Configuring Security for VPNs with IPsec " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Configuring basic AAA RADIUS	" Configuring RADIUS " module in the *Cisco IOS XE Security Configuration Guide: Securing User Services*
Configuring ISAKMP profiles	" VRF-Aware IPsec " module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Privilege levels with TACACS+ and RADIUS	• " Configuring TACACS+ " module in the *Cisco IOS XE Security Configuration Guide: Securing User Services* • " Configuring RADIUS " module in the *Cisco IOS XE Security Configuration Guide: Securing User Services*
IP security, RADIUS, and AAA commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
None.	--

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None.	

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec VPN Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 256: Feature Information for IPsec VPN Accounting

Feature Name	Releases	Feature Information
IPsec VPN Accounting	Cisco IOS XE Release 2.1	The IPsec VPN Accounting feature allows for a session to be accounted for by indicating when the session starts and when it stops. A VPN session is defined as an IKE SA and the one or more SA pairs that are created by the IKE SA. The session starts when the first IPsec pair is created and stops when all IPsec SAs are deleted. Session identifying information and session usage information is passed to the RADIUS server through standard RADIUS attributes and VSAs. The following commands were introduced or modified: **client authentication list, client configuration address, crypto isakmp profile, crypto map (global IPsec), debug crypto isakmp, isakmp authorization list, match identity, set isakmp-profile, vrf**.

Glossary

IKE --Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IP security [IPsec]) that require keys. Before any IPsec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a certification authority (CA) service.

IPsec --IP security. IPsec is A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISAKMP --Internet Security Association and Key Management Protocol. ISAKMP is an Internet IPsec protocol (RFC 2408) that negotiates, establishes, modifies, and deletes security associations. It also exchanges key generation and authentication data (independent of the details of any specific key generation technique), key establishment protocol, encryption algorithm, or authentication mechanism.

L2TP session --Layer 2 Transport Protocol. L2TP are communications transactions between the L2TP access concentrator (LAC) and the L2TP network server (LNS) that support tunneling of a single PPP connection. There is a one-to-one relationship among the PPP connection, L2TP session, and L2TP call.

NAS --network access server. A NAS is a Cisco platform (or collection of platforms, such as an AccessPath system) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, the public switched telephone network [PSTN]).

PFS --perfect forward secrecy. PFS is a cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised because subsequent keys are not derived from previous keys.

QM --Queue Manager. The Cisco IP Queue Manager (IP QM) is an intelligent, IP-based, call-treatment and routing solution that provides powerful call-treatment options as part of the Cisco IP Contact Center (IPCC) solution.

RADIUS --Remote Authentication Dial-In User Service. RADIUS is a database for authenticating modem and ISDN connections and for tracking connection time.

RSA --Rivest, Shamir, and Adelman. Rivest, Shamir, and Adelman are the inventors of the Public-key cryptographic system that can be used for encryption and authentication.

SA --security association. A SA is an instance of security policy and keying material that is applied to a data flow.

TACACS+ --Terminal Access Controller Access Control System Plus. TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server.

VPN --Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF --A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

VSA --vendor-specific attribute. A VSA is an attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

XAUTH --Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

CHAPTER 191

IPsec Usability Enhancements

The IPsec Usability Enhancements feature introduces functionality that eases the configuration and monitoring of your IPsec virtual private network (VPN). Benefits of this feature include intelligent defaults for IPsec and Internet Key Exchange (IKE) and the ability to easily verify and troubleshoot IPsec VPNs.

Prerequisites for IPsec Usability Enhancements

- You must be familiar with IPsec, IKE, and encryption.
- You must have configured IPsec and enabled IKE on your router.
- You must be running Cisco IOS XE k9 crypto image on your router.

Information About IPsec Usability Enhancements

IPsec Overview

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF), which provides security for transmission of sensitive information over public networks. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

IPsec provides secure tunnels between two peers. You may define which packets are considered sensitive and should be sent through these secure tunnels. You may also define the parameters that should be used to protect these sensitive packets by specifying characteristics of the tunnels. When an IPsec peer detects a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

IPsecOperation

An IPsec operation involves five basic steps: identifying interesting traffic, IKE phase-1, IKE phase-2, establishing the tunnel or IPsec session, and finally tearing down the tunnel.

Step 1: Identifying Interesting Traffic

The VPN devices recognize the traffic, or sensitive packets, to detect. IPsec is either applied to the sensitive packet, the packet is bypassed, or the packet is dropped. Based on the traffic type, if IPsec is applied then IKE phase-1 is initiated.

Step 2: IKE Phase-1

There are three exchanges between the VPN devices to negotiate an IKE security policy and establish a secure channel.

During the first exchange, the VPN devices negotiate matching IKE transform sets to protect the IKE exchange resulting in establishing an Internet Security Association and Key Management Protocol (ISAKMP) policy to utilize. The ISAKMP policy consists of an encryption algorithm, a hash algorithm, an authentication algorithm, a Diffie-Hellman (DH) group, and a lifetime parameter.

There are eight default ISAKMP policies supported. For more information on default ISAKMP policies, see the Verifying IKE Phase-1 ISAKMP Default Policies, on page 2475 .

The second exchange consists of a Diffie-Hellman exchange, which establishes a shared secret.

The third exchange authenticates peer identity. After the peers are authenticated, IKE phase-2 begins.

Step 3: IKE Phase-2

The VPN devices negotiate the IPsec security policy used to protect the IPsec data. IPsec transform sets are negotiated.

A transform set is a combination of algorithms and protocols that enact a security policy for network traffic. For more information on default transform sets, see the Verifying Default IPsec Transform-Sets, on page 2478. A VPN tunnel is ready to be established.

Step 4: Establishing the Tunnel--IPsec Session

The VPN devices apply security services to IPsec traffic and then transmit the IPsec data. Security associations (SAs) are exchanged between peers. The negotiated security services are applied to the tunnel traffic while the IPsec session is active.

Step 5: Terminating the Tunnel

The tunnel is torn down when an IPsec SA lifetime time-out occurs or if the packet counter is exceeded. The IPsec SA is removed.

How to Utilize IPsec Usability Enhancements

Verifying IKE Phase-1 ISAKMP Default Policies

When IKE negotiation begins, the peers try to find a common policy, starting with the highest priority policy as specified on the remote peer. The peers negotiate the policy sets until there is a match. If peers have more than one policy set in common, the lowest priority number is used.

There are three groups of IKE phase-1, ISAKMP, policies as defined by policy priority ranges and behavior:

- Default ISAKMP policies, which are automatically enabled.
- User configured ISAKMP policies, which you may configure with the **crypto isakmp policy** command.
- Easy VPN ISAKMP policies, which are made available during Easy VPN configuration.

This section describes the three groups of ISAKMP policies, how they behave in relationship to one another, how to determine which policies are in use with the appropriate **show** command, and how to disable the default ISAKMP policies.

Default IKE Phase-1 Policies

There are eight default IKE phase-1, ISAKMP, policies supported (see the table below) that are enabled automatically. If you have neither manually configured IKE policies with the **crypto isakmp policy** command nor disabled the default IKE policies with the **no crypto isakmp default policy** command, the default IKE policies will be used during peer IKE negotiations. You can verify that the default IKE policies are in use by issuing either the **show crypto isakmp policy** command or the **show crypto isakmp default policy** command.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The default IKE policies define the following policy set parameters:

- The priority, 65507-65514, where 65507 is the highest priority and 65514 is the lowest priority.
- The authentication method, Rivest, Shamir, and Adelman (RSA) or preshared keys (PSK).
- The encryption method, Advanced Encryption Standard (AES) or Triple Data Encryption Standard (3DES).
- The hash function, Secure Hash Algorithm (SHA-1) or Message-Digest algorithm 5 (MD5).
- The DH group specification DH2 or DH5
  - DH2 specifies the 768-bit DH group.
  - DH5 specifies the 1536-bit DH group.

Note Cisco no longer recommends using 3DES, MD5 and DH groups 1, 2 and 5. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper. To learn more about IKE configuration, read the chapter "Configuring Internet Key Exchange for IPsec VPNs" in *Internet Key Exchange for IPsec VPNs Configuration Guide*.

Table 257: Default IKE Phase-1, ISAKMP, Policies

Priority	Authentication	Encryption	Hash	Diffie-Hellman
65507	RSA	AES	SHA	DH5
65508	PSK	AES	SHA	DH5
65509	RSA	AES	MD5	DH5
65510	PSK	AES	MD5	DH5
65511	RSA	3DES	SHA	DH2
65512	PSK	3DES	SHA	DH2
65513	RSA	3DES	MD5	DH2
65514	PSK	3DES	MD5	DH2

User Configured IKE Policies

You may configure IKE policies with the **crypto isakmp policy** command. User configured IKE policies are uniquely identified and configured with a priority number ranging from 1-10000, where 1 is the highest priority and 10000 the lowest priority.

Once you have configured one or more IKE policies with a priority of 1-10000:

- The user configured policies will be used during peer IKE negotiations.
- The default IKE policies will no longer used during peer IKE negotiations.
- The user configured policies may be displayed by issuing the **show crypto isakmp policy** command.

Easy VPN ISAKMP Policies

If you have configured Easy VPN, the default Easy VPN ISAKMP policies in use are uniquely identified with a priority number ranging from 65515-65535, where 65515 is the highest priority and 65535 is the lowest priority.

Once a user has configured Easy VPN:

- The default Easy VPN ISAKMP policies and the default IKE policies will be used during peer IKE negotiations.
- The Easy VPN ISAKMP policies and the default IKE policies will be displayed by issuing the **show crypto isakmp policy** command.

- Default ISAKMP policies will be displayed by issuing the **show crypto isakmp default policy** command unless they have been disabled by issuing the **no crypto isakmp default policy** command.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp default policy**
3. **configure terminal**
4. **no crypto isakmp default policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp default policy** **Example:** `Router# show crypto isakmp default policy`	(Optional) Displays default ISAKMP policies if no policy with a priority of 1-10000 is configured.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**no crypto isakmp default policy** **Example:** `Router(config)# no crypto isakmp default policy`	(Optional) Turns off default ISAKMP policies with priorities 65507-65514.

Examples

The following is sample output of the **show crypto isakmp default policy** command. The default policies are displayed because the default policies have not been disabled.

```
Router# show crypto isakmp default policy

Default IKE policy
Default protection suite of priority 65507
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65508
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Secure Hash Standard
```

```
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65509
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65510
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65511
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65512
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65513
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65514
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
```

The following example disables the default IKE policies then shows the resulting output of the **show crypto isakmp default policy** command, which is blank:

```
Router# configure terminal
Router(config)# no crypto isakmp default policy
Router(config)# exit
Router# show crypto isakmp default policy
Router#
!There is no output since the default IKE policies have been disabled.
```

The following is an example system log message that is generated whenever the default ISAKMP policies are in use:

```
%CRYPTO-6-IKMP_POLICY_DEFAULT: Using ISAKMP Default policies
```

Verifying Default IPsec Transform-Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

During IPsec SA negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and is applied to the protected traffic as part of the IPsec SAs of both peers.

Default Transform Sets

A default transform set will be used by any crypto map or IPsec profile where no other transform set has been configured and if the following is true:

- The default transform sets have not been disabled with the **no crypto ipsec default transform-set** command.
- The crypto engine in use supports the encryption algorithm.

The two default transform sets each define an Encapsulation Security Protocol (ESP) encryption transform type and an ESP authentication transform type as shown in the table below.

Table 258: Default Transform Sets and Parameters

Default Transform Name	ESP Encryption Transform and Description	ESP Authentication Transform and Description
#$!default_transform_set_0	esp-3des (ESP with the 168-bit 3DES or Triple DES encryption algorithm)	esp-sha-hmac
#$!default_transform_set_1	esp-aes (ESP with the 128-bit AES encryption algorithm)	esp-sha-hmac (ESP with the SHA-1, hash message authentication code [HMAC] variant authentication algorithm)

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec default transform-set**
3. **configure terminal**
4. **no crypto ipsec default transform-set**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec default transform-set** **Example:** `Router# show crypto ipsec default transform-set`	(Optional) Displays the default IPsec transform sets currently in use by IKE.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**no crypto ipsec default transform-set** **Example:** `Router(config)# no crypto ipsec default transform-set`	(Optional) Disables the default IPsec transform sets.

Examples

```
The following example displays output from the show crypto ipsec default transform-set
 command when the default transform sets are enabled, the default setting:
Router# show crypto ipsec default transform-set
Transform set #$!default_transform_set_1: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set #$!default_transform_set_0: { esp-3des esp-sha-hmac  }
   will negotiate = { Transport,  },
```

The following example displays output from the **show crypto ipsec default transform-set** command when the default transform sets have been disabled with the **no crypto ipsec default transform-set**command.

```
Router(config)# no crypto ipsec default transform-set
Router(config)# exit
Router#
Router# show crypto ipsec default transform-set
! There is no output.
Router#
```

The following is an example system log message that is generated whenever IPsec SAs have negotiated with a default transform set:

```
%CRYPTO-5-IPSEC_DEFAULT_TRANSFORM: Using Default IPsec transform-set
```

Verifying and Troubleshooting IPsec VPNs

Perform one of the following optional tasks in this section, depending on whether you want to verify IKE phase-1 or IKE phase-2 tunnels or troubleshoot your IPsec VPN:

Verifying IKE Phase-1 ISAKMP

To display statistics for ISAKMP tunnels, use the following optional commands.

SUMMARY STEPS

1. **show crypto mib isakmp flowmib failure [vrf** *vrf-name* **]**

2. **show crypto mib isakmp flowmib global [vrf** *vrf-name* **]**
3. **show crypto mib isakmp flowmib history [vrf** *vrf-name* **]**
4. **show crypto mib isakmp flowmib peer [index** *peer-mib-index* **] [vrf** *vrf-name* **]**
5. **show crypto mib isakmp flowmib tunnel [index** *tunnel-mib-index* **] [vrf** *vrf-name* **]**

DETAILED STEPS

Step 1 **show crypto mib isakmp flowmib failure [vrf** *vrf-name* **]**

For ISAKMP tunnel failures, this command displays event information. The following is sample output for this command:

Example:

```
Router# show crypto mib isakmp flowmib failure
 vrf Global
  Index:                       1
  Reason:                      peer lost
  Failure time since reset:    00:07:27
  Local type:                  ID_IPV4_ADDR
  Local value:                192.0.2.1
  Remote type:                 ID_IPV4_ADDR
  Remote Value:                192.0.2.2
  Local Address:               192.0.2.1
  Remote Address:              192.0.2.2
  Index:                       2
  Reason:                      peer lost
  Failure time since reset:    00:07:27
  Local type:                  ID_IPV4_ADDR
  Local value:                192.0.3.1
  Remote type:                 ID_IPV4_ADDR
  Remote Value:                192.0.3.2
  Local Address:               192.0.3.1
  Remote Address:              192.0.3.2
  Index:                       3
  Reason:                      peer lost
  Failure time since reset:    00:07:32
  Local type:                  ID_IPV4_ADDR
  Remote type:                 ID_IPV4_ADDR
  Remote Value:                192.0.2.2
  Local Address:               192.0.2.1
  Remote Address:              192.0.2.2
```

Step 2 **show crypto mib isakmp flowmib global [vrf** *vrf-name* **]**

Global ISAKMP tunnel statistics are displayed by issuing this command. The following is sample output for this command:

Example:

```
Router# show crypto mib isakmp flowmib global
vrf Global
  Active Tunnels:                          3
  Previous Tunnels:                        0
  In octets:                               2856
  Out octets:                              3396
  In packets:                              16
  Out packets:                             19
  In packets drop:                         0
  Out packets drop:                        0
  In notifys:                              4
  Out notifys:                             7
```

```
  In P2 exchg:                          3
  Out P2 exchg:                         6
  In P2 exchg invalids:                 0
  Out P2 exchg invalids:                0
  In P2 exchg rejects:                  0
  Out P2 exchg rejects:                 0
  In IPSEC delete:                      0
  Out IPSEC delete:                     0
  SAs locally initiated:                3
  SAs locally initiated failed:         0
  SAs remotely initiated failed:        0
  System capacity failures:             0
  Authentication failures:              0
  Decrypt failures:                     0
  Hash failures:                        0
  Invalid SPI:                          0
```

Step 3 **show crypto mib isakmp flowmib history [vrf** *vrf-name* **]**

For information about ISAKMP tunnels that are no longer active, this command displays event information including the reason that the tunnel was terminated. The following is sample output for this command:

Example:

```
Router# show crypto mib isakmp flowmib history
vrf Global
  Reason:                           peer lost
  Index:                            2
  Local type:                       ID_IPV4_ADDR
  Local address:                    192.0.2.1
  Remote type:                      ID_IPV4_ADDR
  Remote address:                   192.0.2.2
  Negotiation mode:                 Main Mode
  Diffie Hellman Grp:               2
  Encryption algo:                  des
  Hash algo:                        sha
  Auth method:                      psk
  Lifetime:                         86400
  Active time:                      00:06:30
  Policy priority:                  1
  Keepalive enabled:                Yes
  In octets:                        3024
  In packets:                       22
  In drops:                         0
  In notifys:                       18
  In P2 exchanges:                  1
  In P2 exchg invalids:             0
  In P2 exchg rejected:               0
  In P2 SA delete reqs:               0
  Out octets:                         4188
  Out packets:                        33
  Out drops:                          0
  Out notifys:                        28
  Out P2 exchgs:                      2
  Out P2 exchg invalids:              0
  Out P2 exchg rejects:               0
  Out P2 Sa delete requests:          0
  Reason:                             peer lost
  Index:                              3
  Local type:                         ID_IPV4_ADDR
  Local address:                      192.0.3.1
  Remote type:                        ID_IPV4_ADDR
  Remote address:                     192.0.3.2
  Negotiation mode:                   Main Mode
```

```
  Diffie Hellman Grp:                  2
  Encryption algo:                     des
  Hash algo:                           sha
  Auth method:                         psk
  Lifetime:                            86400
  Active time:                         00:06:25
  Policy priority:                     1
  Keepalive enabled:                   Yes
  In octets:                           3140
  In packets:                          23
  In drops:                            0
  In notifys:                          19
  In P2 exchanges:                     1
  In P2 exchg invalids:                0
  In P2 exchg rejected:                0
  In P2 SA delete reqs:                0
  Out octets:                          4304
  Out packets:                         34
  Out drops:                           0
  Out notifys:                         29
  Out P2 exchgs:                       2
  Out P2 exchg invalids:               0
  Out P2 exchg rejects:                0
  Out P2 Sa delete requests:           0
```

Step 4 **show crypto mib isakmp flowmib peer [index** *peer-mib-index* **] [vrf** *vrf-name* **]**

For active ISAKMP peer associations, this command displays information including indexes, type of connection, and IP addresses. The following is sample output for this command:

Example:

```
Router# show crypto mib isakmp flowmib peer
vrf Global
  Index:              1
  Local type:         ID_IPV4_ADDR
  Local address:      192.0.2.1
  Remote type:        ID_IPV4_ADDR
  Remote address:     192.0.2.2
  Index:              2
  Local type:         ID_IPV4_ADDR
  Local address:      192.0.3.1
  Remote type:        ID_IPV4_ADDR
  Remote address:     192.0.3.1
  Index:              3
  Local type:         ID_IPV4_ADDR
  Local address:      192.0.4.1
  Remote type:        ID_IPV4_ADDR
  Remote address:     192.0.4.1
```

Step 5 **show crypto mib isakmp flowmib tunnel [index** *tunnel-mib-index* **] [vrf** *vrf-name* **]**

For active ISAKMP tunnels, this command displays tunnel statistics. The following is sample output for this command:

Example:

```
Router# show crypto mib isakmp flowmib tunnel
vrf Global
  Index:                    1
  Local type:               ID_IPV4_ADDR
  Local address:            192.0.2.1
  Remote type:              ID_IPV4_ADDR
  Remote address:           192.0.2.2
```

```
  Negotiation mode:           Main Mode
  Diffie Hellman Grp:         2
  Encryption algo:            des
  Hash algo:                  sha
  Auth method:                psk
  Lifetime:                   86400
  Active time:                00:03:08
  Policy priority:            1
  Keepalive enabled:          Yes
  In octets:                  2148
  In packets:                 15
  In drops:                   0
  In notifys:                 11
  In P2 exchanges:            1
  In P2 exchg invalids:       0
  In P2 exchg rejected:       0
  In P2 SA delete reqs:       0
  Out octets:                 2328
  Out packets:                16
  Out drops:                  0
  Out notifys:                12
  Out P2 exchgs:              2
  Out P2 exchg invalids:      0
  Out P2 exchg rejects:       0
  Out P2 Sa delete requests:  0
```

Verifying IKE Phase-2

To display statistics for IPsec phase-2 tunnels, use the following optional commands.

SUMMARY STEPS

1. **show crypto mib ipsec flowmib endpoint [vrf** *vrf-name* **]**
2. **show crypto mib ipsec flowmib failure [vrf** *vrf-name* **]**
3. **show crypto mib ipsec flowmib global [vrf** *vrf-name* **]**
4. **show crypto mib ipsec flowmib history [vrf** *vrf-name* **]**
5. **show crypto mib ipsec flowmib spi [vrf** *vrf-name* **]**
6. **show crypto mib ipsec flowmib tunnel** [**index** *tunnel-mib-index*] **[vrf** *vrf-name* **]**

DETAILED STEPS

Step 1 **show crypto mib ipsec flowmib endpoint [vrf** *vrf-name* **]**

Information for each active endpoint, local or remote device, associated with an IPsec phase-2 tunnel is displayed by issuing this command. The following is sample output for this command:

Example:

```
Router#  show crypto mib ipsec flowmib endpoint
vrf Global
  Index:               1
  Local type:          Single IP address
  Local address:       192.1.2.1
  Protocol:            0
  Local port:          0
```

```
  Remote type:         Single IP address
  Remote address:      192.1.2.2
  Remote port:         0
  Index:               2
  Local type:          Subnet
  Local address:       192.1.3.0 255.255.255.0
  Protocol:            0
  Local port:          0
  Remote type:         Subnet
  Remote address:      192.1.3.0 255.255.255.0
  Remote port:         0
```

Step 2 **show crypto mib ipsec flowmib failure [vrf** *vrf-name* **]**

For ISAKMP tunnel failures, this command displays event information. The following is sample output for this command:

Example:

```
Router# show crypto mib ipsec flowmib failure
vrf Global
  Index:                        1
  Reason:                       Operation request
  Failure time since reset:     00:25:18
  Src address:                  192.1.2.1
  Destination address:          192.1.2.2
  SPI:                          0
```

Step 3 **show crypto mib ipsec flowmib global [vrf** *vrf-name* **]**

Global IKE phase-2 tunnel statistics are displayed by issuing this command. The following is sample output for this command:

Example:

```
Router# show crypto mib ipsec flowmib global
 vrf Global
  Active Tunnels:                    2
  Previous Tunnels:                  0
  In octets:                         800
  Out octets:                        1408
  In packets:                        8
  Out packets:                       8
  Uncompressed encrypted bytes:    1408
  In packets drops:                  0
  Out packets drops:                 2
  In replay drops:                   0
  In authentications:                8
  Out authentications:               8
  In decrypts:                       8
  Out encrypts:                      8
  Compressed bytes:                  0
  Uncompressed bytes:                0
  In uncompressed bytes:             0
  Out uncompressed bytes:            0
  In decrypt failures:               0
  Out encrypt failures:              0
  No SA failures:                    0
! Number of SA Failures.
  Protocol use failures:             0
  System capacity failures:          0
  In authentication failures:        0
  Out authentication failures:       0
```

Step 4 **show crypto mib ipsec flowmib history [vrf** *vrf-name* **]**

For information about IKE phase-2 tunnels that are no longer active, this command displays event information including the reason that the tunnel was terminated. The following is sample output for this command:

Example:

```
Router# show crypto mib ipsec flowmib history
vrf Global
  Reason:                      Operation request
  Index:                       1
  Local address:               192.1.2.1
  Remote address:              192.1.2.2
  IPSEC keying:                IKE
  Encapsulation mode:          1
  Lifetime (KB):               4608000
  Lifetime (Sec):              3600
  Active time:                 00:24:32
  Lifetime threshold (KB):     423559168
  Lifetime threshold (Sec):    3590000
  Total number of refreshes:   0
  Expired SA instances:        4
  Current SA instances:        4
  In SA DH group:              14
  In sa encrypt algorithm      aes
  In SA auth algorithm:        rsig
  In SA ESP auth algo:         ESP_HMAC_SHA
  In SA uncompress algorithm:  None
  Out SA DH group:             14
  Out SA encryption algorithm: aes
  Out SA auth algorithm:       ESP_HMAC_SHA
  Out SA ESP auth algorithm:   ESP_HMAC_SHA
  Out SA uncompress algorithm: None
  In octets:                   400
  Decompressed octets:         400
  In packets:                  4
  In drops:                        0
  In replay drops:                 0
  In authentications:              4
  In authentication failures:      0
  In decrypts:                     4
  In decrypt failures:             0
  Out octets:                      704
  Out uncompressed octets:         704
  Out packets:                     4
  Out drops:                       1
  Out authentications:             4
  Out authentication failures:     0
  Out encryptions:                 4
  Out encryption failures:         0
  Compressed octets:               0
  Decompressed octets:             0
  Out uncompressed octets:         704
```

Step 5 **show crypto mib ipsec flowmib spi [vrf** *vrf-name* **]**

The security protection index (SPI) table contains an entry for each active and expiring security IKE phase-2 association. The following is sample output for this command, which displays the SPI table:

Example:

```
Router# show crypto mib ipsec flowmib spi
vrf Global
```

```
  Tunnel Index:          1
  SPI Index:             1
  SPI Value:             0xCC57D053
  SPI Direction:         In
  SPI Protocol:          AH
  SPI Status:            Active
  SPI Index:             2
  SPI Value:             0x68612DF
  SPI Direction:         Out
  SPI Protocol:          AH
  SPI Status:            Active
  SPI Index:             3
  SPI Value:             0x56947526
  SPI Direction:         In
  SPI Protocol:          ESP
  SPI Status:            Active
  SPI Index:             4
  SPI Value:             0x8D7C2204
  SPI Direction:         Out
  SPI Protocol:          ESP
  SPI Status:            Active
```

Step 6 **show crypto mib ipsec flowmib tunnel** [**index** *tunnel-mib-index*] **[vrf** *vrf-name* **]**

For active IKE phase-2 tunnels, this command displays tunnel statistics. The following is sample output for this command:

Example:

```
Router# show crypto mib ipsec flowmib tunnel
vrf Global
  Index:                            1
  Local address:                    192.0.2.1
  Remote address:                   192.0.2.2
  IPSEC keying:                     IKE
  Encapsulation mode:               1
  Lifetime (KB):                    4608000
  Lifetime (Sec):                   3600
  Active time:                      00:05:46
  Lifetime threshold (KB):          64
  Lifetime threshold (Sec):         10
  Total number of refreshes:        0
  Expired SA instances:             0
  Current SA instances:             4
  In SA DH group:                   14
  In sa encrypt algorithm:          aes
  In SA auth algorithm:             rsig
  In SA ESP auth algo:              ESP_HMAC_SHA
  In SA uncompress algorithm:       None
  Out SA DH group:                  14
  Out SA encryption algorithm:      aes
  Out SA auth algorithm:            ESP_HMAC_SHA
  Out SA ESP auth algorithm:        ESP_HMAC_SHA
  Out SA uncompress algorithm:      None
  In octets:                           400
  Decompressed octets:                 400
  In packets:                          4
  In drops:                            0
  In replay drops:                     0
  In authentications:                  4
  In authentication failures:          0
  In decrypts:                         4
  In decrypt failures:                 0
  Out octets:                          704
  Out uncompressed octets:             704
```

```
Out packets:                          4
Out drops:                            1
Out authentications:                  4
Out authentication failures:          0
Out encryptions:                      4
Out encryption failures:              0
Compressed octets:                    0
Decompressed octets:                  0
Out uncompressed octets:              704
```

Troubleshooting IPsec VPNs

The **show tech-support ipsec** command simplifies the collection of the IPsec related information if you are troubleshooting a problem.

SUMMARY STEPS

1. **show tech-support ipsec**

DETAILED STEPS

show tech-support ipsec

There are three variations of the **show tech-support ipsec**command:

- **show tech-support ipsec**
- **show tech-support ipsec peer** *ipv4address*
- **show tech-support ipsec vrf** *vrf-name*

For a sample display of the output from the **show tech-support ipsec** command for the individual **show** commands listed below for each variation see the following sections.

Output of the show tech-support ipsec Command

If you enter the **show tech-support ipsec**command without any keywords, the command output displays the following **show** commands, in order of output:

- **show version**
- **show running-config**
- **show crypto isakmp sa count**
- **show crypto ipsec sa count**
- **show crypto session summary**
- **show crypto session detail**
- **show crypto isakmp sa detail**
- **show crypto ipsec sa detail**
- **show crypto isakmp peers**

- **show crypto ruleset detail**
- **show processes memory | include Crypto IKMP**
- **show processes cpu | include Crypto IKMP**
- **show crypto eli**
- **show crypto engine accelerator statistic**

Output of the show tech-support ipsec peer Command

If you enter the **show tech-support ipsec**command with the **peer** keyword and the *ipv4address* argument, the output displays the following **show** commands, in order of output for the specified peer:

- **show version**
- **show running-config**
- **show crypto session remote** *ipv4address* **detail**
- **show crypto isakmp sa peer** *ipv4address* **detail**
- **show crypto ipsec sa peer** *ipv4address* **detail**
- **show crypto isakmp peers** *ipv4address*
- **show crypto ruleset detail**
- **show processes memory | include Crypto IKMP**
- **show processes cpu | include Crypto IKMP**
- **show crypto eli**
- **show crypto engine accelerator statistic**

Output of the show tech-support ipsec vrf Command

If you enter the **show tech-support ipsec**command with the **vrf** keyword and the *vrf-name*argument, the output displays the following **show** commands, in order of output for the specified Virtual Routing and Forwarding (VRF):

- **show version**
- **show running-config**
- **show crypto isakmp sa count vrf** *vrf-name*
- **show crypto ipsec sa count vrf** *vrf-name*
- **show crypto session ivrf** *ivrf-name* **detail**
- **show crypto session fvrf** *fvrf-name* **detail**
- **show crypto isakmp sa vrf** *vrf-name* **detail**
- **show crypto ipsec sa vrf** *vrf-name* **detail**
- **show crypto ruleset detail**
- **show processes memory | include Crypto IKMP**

- **show processes cpu | include Crypto IKMP**
- **show crypto eli**
- **show crypto engine accelerator statistic**

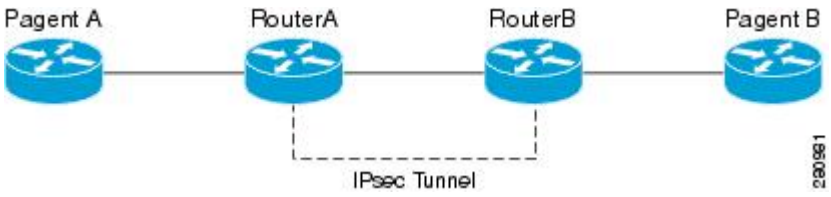
Example:

Configuration Examples for IPsec Usability Enhancements

IKE Default Policies Example

In the following example, crypto maps are configured on RouterA and RouterB and default IKE policies are in use. Traffic is routed from Pagent A to Pagent B. Checking the system log on Peer A and Peer B confirms that the default IKE policies are in use on both peers (see the figure below).

Figure 96: Example Site to Site Topology




```
! Configuring RouterA.
RouterA(config)# crypto isakmp key identity address 209.165.200.226
RouterA(config)# crypto map testmap 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
 and a valid access list have been configured.
RouterA(config-crypto-map)# set peer 209.165.200.226
RouterA(config-crypto-map)# match address 101
RouterA(config-crypto-map)# exit
RouterA(config)# ip route 209.165.200.225 255.255.255.224 209.165.200.226
RouterA(config)# access-list 101 permit ip host 209.165.200.227 host 209.165.200.225
RouterA(config)# end
RouterA(config)# interface FastEthernet1/2
RouterA(config-if)# crypto map testmap
RouterA(config-if)# end
RouterA(config)# crypto ipsec transform test_transf esp-aes esp-sha-hmac
RouterA(cfg-crypto-trans)# mode tunnel
RouterA(cfg-crypto-trans)# end
RouterA(config)# crypto map testmap 10
RouterA(config-crypto-map)# set transform-set test_transf
RouterA(config-crypto-map)# end
! Configuring RouterB.
RouterB(config)# crypto isakmp key identity address 209.165.200.228
RouterB(config)# crypto dynamic-map dyn_testmap 10
RouterB(config-crypto-map)# crypto map testmap 10 ipsec-isakmp dynamic dyn_testmap
RouterB(config)# ip route 209.165.200.227 255.255.255.224 209.165.200.228
RouterB(config)# end
RouterB(config)# interface GigabitEthernet0/1
```

```
RouterB(config-if)# crypto map testmap
RouterB(config-if)# end
RouterB(config)# crypto ipsec transform test_transf esp-aes esp-sha-hmac
RouterB(cfg-crypto-trans)# mode tunnel
RouterB(cfg-crypto-trans)# end
RouterB(config)# crypto dynamic-map dyn_testmap 10
RouterB(config-crypto-map)# set transform-set test_transf
RouterB(config-crypto-map)# end
! Routing traffic from PagentA to PagentB.
PagentA(config)# ip route 209.165.200.225 255.255.255.224 209.165.200.229
PagentA(config)# end
! Routing traffic from PagentB to PagentA.
PagentB(config)# ip route 209.165.200.227 255.255.255.224 209.165.200.230
PagentB(config)# end
! Checking the system log on RouterA confirms that the default IKE policies are in use.
RouterA# show log | include %CRYPTO-6-IKMP_POLICY_DEFAULT*
Jun  5 09:17:59.251 PDT: %CRYPTO-6-IKMP_POLICY_DEFAULT: Using ISAKMP Default policies
! Checking the system log on RouterB confirms that the default IKE policies are in use.
RouterB# show log | include %CRYPTO-6-IKMP_POLICY_DEFAULT*
Jun  5 09:17:59.979 PDT: %CRYPTO-6-IKMP_POLICY_DEFAULT: Using ISAKMP Default policies
```

Default Transform Sets Example

In the following example, static crypto maps are configured on RouterA and dynamic crypto maps are configured on RouterB. Traffic is routed from Pagent A to Pagent B. The IPsec SAs negotiate with default transform sets and the traffic is encrypted. Executing the **show crypto map** command on both peers verifies that the default transform sets are in use.

```
! Configuring RouterA.
RouterA(config)# crypto isakmp key identify address 209.165.200.225
RouterA(config)# crypto map testmap 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
 and a valid access list have been configured.
RouterA(config-crypto-map)# set peer 209.165.200.225
RouterA(config-crypto-map)# match address 101
RouterA(config-crypto-map)# exit
RouterA(config)# ip route 209.165.200.226 255.255.255.255 209.165.200.225
RouterA(config)# access-list 101 permit ip host 209.165.200.227 host 209.165.200.226
RouterA(config)# end
RouterA(config)# interface FastEthernet1/2
RouterA(config-if)# crypto map testmap
RouterA(config-if)# end
RouterA(config)# crypto isakmp policy 10
RouterA(config-isakmp)# encryption aes
RouterA(config-isakmp)# authentication pre-share
RouterA(config-isakmp)# hash sha
RouterA(config-isakmp)# group 5
RouterA(config-isakmp)# end
! Configuring RouterB.
RouterB(config)# crypto isakmp key identity address 209.165.200.229
RouterB(config)# crypto dynamic-map dyn_testmap 10
RouterB(config-crypto-map)# crypto map testmap 10 ipsec-isakmp dynamic dyn_testmap
RouterB(config)# ip route 209.165.200.227 255.255.255.255 209.165.200.229
RouterB(config)# end
RouterB(config)# interface GigabitEthernet0/1
RouterB(config-if)# crypto map testmap
RouterB(config-if)# end
RouterB(config)# crypto isakmp policy 10
RouterB(config-isakmp)# encryption aes
RouterB(config-isakmp)# authentication pre-share
```

```
RouterB(config-isakmp)# hash sha
RouterB(config-isakmp)# group 5
RouterB(config-isakmp)# end
! The SA is using the default transform set and traffic is encrypted on RouterA.
RouterA# show crypto isakmp sa detail | include 209.165.200.229.*209.165.200.225.*ACTIVE
13007  209.165.200.229     209.165.200.225     ACTIVE aes  sha  psk  5  23:59:56
13006  209.165.200.229     209.165.200.225     ACTIVE aes  sha  psk  5  0
13005  209.165.200.229     209.165.200.225     ACTIVE aes  sha  psk  5  0
! The SA is using the default transform set and traffic is encrypted on RouterB.
RouterB# show crypto isakmp sa detail | include 209.165.200.225.*209.165.200.229.*ACTIVE
7007  209.165.200.225      209.165.200.229      ACTIVE aes  sha  psk  5  23:59:55
7006  209.165.200.225      209.165.200.229      ACTIVE aes  sha  psk  5  0
7005  209.165.200.225      209.165.200.229      ACTIVE aes  sha  psk  5  0
! Verifying that the default transform sets are in use on RouterA.
RouterA# show crypto map
Crypto Map "testmap" 10 ipsec-isakmp
 Peer = 209.165.200.225
 Extended IP access list 101
     access-list 101 permit ip host 209.165.200.227 host 209.165.200.226
 Current peer: 209.165.200.225
 Security association lifetime: 4608000 kilobytes/3600 seconds
 PFS (Y/N): N
 Transform sets={
  #$!default_transform_set_1:  { esp-aes esp-sha-hmac  } ,
  #$!default_transform_set_0:  { esp-3des esp-sha-hmac  } ,
 }
 Interfaces using crypto map testmap:
  FastEthernet1/2
! Verifying that the default transform sets are in use on RouterB.
RouterB# show crypto map
Crypto Map "testmap" 10 ipsec-isakmp
 Dynamic map template tag: dyn_testmap
Crypto Map "testmap" 65536 ipsec-isakmp
 Peer = 209.165.200.229
 Extended IP access list
     access-list permit ip host 209.165.200.226 host 209.165.200.227
     dynamic (created from dynamic map dyn_testmap/10)
 Current peer: 209.165.200.229
 Security association lifetime: 4608000 kilobytes/3600 seconds
 PFS (Y/N): N
 Transform sets={
  #$!default_transform_set_1:  { esp-aes esp-sha-hmac  } ,
 }
 Interfaces using crypto map testmap:
  GigabitEthernet0/1
```

Additional References

The following sections provide references related to the IPsec Usability Enhancement feature.

Related Documents

Related Topic	Document Title
IKE configuration	Configuring Internet Key Exchange for IPsec VPNs module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
IPsec configuration	Configuring Security for VPNs with IPsec module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*

Related Topic	Document Title
Easy VPN server	Easy VPN Server module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*
Cisco IOS XE security commands	*Cisco IOS Security Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IPsec Usability Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 259: Feature Information for IPsec Usability Enhancements

Feature Name	Releases	Feature Information
IPsec Usability Enhancements	Cisco IOS XE Release 2.4	This feature introduces intelligent defaults for IKE and IPsec, and **show** commands to access MIB statistics and to aid in troubleshooting. The following commands were introduced or modified: **crypto ipsec default transform-set**, **crypto isakmp default policy**, **crypto isakmp policy**, **show crypto ipsec default transform-set**, **show crypto ipsec transform-set**, **show crypto isakmp default policy**, **show crypto isakmp policy**, **show crypto map (IPsec)**, **show crypto mib ipsec flowmib endpoint**, **show crypto mib ipsec flowmib failure**, **show crypto mib ipsec flowmib global**, **show crypto mib ipsec flowmib history**, **show crypto mib ipsec flowmib spi**, **show crypto mib ipsec flowmib tunnel**, **show crypto mib isakmp flowmib failure**, **show crypto mib isakmp flowmib global**, **show crypto mib isakmp flowmib history**, **show crypto mib isakmp flowmib peer**, **show crypto mib isakmp flowmib tunnel**, **show tech-support ipsec**.

Glossary

peer--In the context of this module, a router or other device that participates in IPsec.

SA--security association. Description of how two or more entities use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

transform--List of operations performed on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

tunnel--In the context of this module, a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.

PART XIX

VPN Availability

CHAPTER 192

Reverse Route Injection

Reverse route injection (RRI) is the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted.

Prerequisites for Reverse Route Injection

- IP routing should be enabled and static routes should be redistributed if dynamic routing protocols are to be used to propagate RRI-generated static routes.

Restrictions for Reverse Route Injection

- For static crypto maps, routes are always present if RRI is configured on an applied crypto map. The default behavior--of routes always being present for a static map--will not apply unless the **static** keyword is added to the **reverse-route** command.
- Suppose that for a prefix in the RIB we have a manually configured static route with a tag and a route without a tag inserted through RRI. In such a scenario, the route selection may be inconsistent, and either the manually configured route or the RRI route may be chosen.

  To prevent such an inconsistency, perform one of the following actions:

  - If you are manually configuring static routes to all the peer VPN networks of the router, disable RRI by removing reverse route configuration from the crypto map.

- Set an identical tag in the crypto map for the route inserted through RRI.

Information About Reverse Route Injection

Reverse Route Injection

RRI is the ability for static routes to be automatically inserted into the routing process for those networks and hosts that are protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote VPN router as the next hop, the traffic is forced through the crypto process to be encrypted.

After the static route is created on the VPN router, this information is propagated to upstream devices, allowing them to determine the appropriate VPN router to which to send returning traffic in order to maintain IPsec state flows. Being able to determine the appropriate VPN router is particularly useful if multiple VPN routers are used at a site to provide load balancing or failover or if the remote VPN devices are not accessible via a default route. Routes are created in either the global routing table or the appropriate virtual route forwarding (VRF) table.

RRI is applied on a per-crypto map basis, whether this is via a static crypto map or a dynamic crypto map template. The default behavior for the two map types is as follows:

- In the case of a dynamic crypto map, routes are created upon the successful establishment of IPsec security associations (SAs) for those remote proxies. The next hop back to those remote proxies is via the remote VPN router whose address is learned and applied during the creation of the dynamic crypto map template. The routes are deleted after the SAs are deleted. Routes created on the basis of IPsec source proxies on static crypto maps is the default behavior on static maps and overrides the creation of routes on the basis of crypto ACLs (see the next bullet).
- For static crypto maps, routes are created on the basis of the destination information defined in the crypto access list. The next hop is taken from the first set peer statement that is attached to the crypto map. If at any time, RRI, the peer, or the access list is removed from the crypto map, routes will be deleted. This behavior changes with the addition of the RRI enhancements, as explained in the sections below.

How to Configure Reverse Route Injection

Configuring RRI Under a Static Crypto Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map {** *map-name* } **{** *seq-name*} **ipsec-isakmp**
4. **reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer**[**static**] | **remote-peer** *ip-address* [**static**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map {** *map-name* } **{** *seq-name*} **ipsec-isakmp** **Example:** `Router (config)# crypto map mymap 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer**[**static**] | **remote-peer** *ip-address* [**static**]] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuring RRI Under a Dynamic Map Template

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name* *dynamic-seq-name*
4. **reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer**[**static**] | **remote-peer** *ip-address* [**static**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-name* **Example:** `Router (config)# crypto dynamic-map mymap 1`	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
Step 4	**reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer**[**static**] | **remote-peer** *ip-address* [**static**]] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuration Examples for Reverse Route Injection

Configuring RRI When Crypto ACLs Exist Example

The following example shows that all remote VPN gateways connect to the router via 192.168.0.3. RRI is added on the static crypto map, which creates routes on the basis of the source network and source netmask that are defined in the crypto access control list (ACL):

```
crypto map mymap 1 ipsec-isakmp
 set peer 10.1.1.1
 reverse-route
 set transform-set esp-3des-sha
 match address 102
Interface FastEthernet 0/0/1
 ip address 192.168.0.2 255.255.255.0
 standby name group1
 standby ip 192.168.0.3
 crypto map mymap redundancy group1
access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

Configuring RRI When Two Routes Are Created One for the Remote Endpoint and One for Route Recursion Example

In the following example, two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface on which the crypto map is configured:

```
reverse-route remote-peer
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Reverse Route Injection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 260: Feature Information for Reverse Route Injection

Feature Name	Releases	Feature Information
Reverse Route Injection	Cisco IOS XE Release 2.1	Reverse route injection (RRI) is the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities. Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted. The following sections provide information about this feature: The following commands were introduced or modified by this feature: **reverse-route**.

CHAPTER 193

IPsec VPN High Availability Enhancements

The IPsec VPN High Availability Enhancements feature: Reverse Route Injection (RRI) and Hot Standby Router Protocol (HSRP) with IPsec. When used together, these two features provide you with a simplified network design for VPNs and reduced configuration complexity on remote peers when defining gateway lists.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Information About IPsec VPN High Availability Enhancements

Reverse Route Injection

Reverse Route Injection (RRI) simplifies network design for Virtual Private Networks (VPNs) in which there is a requirement for redundancy or load balancing. RRI works with both dynamic and static crypto maps.

RRI provides the following benefits:

- Enables routing of IPsec traffic to a specific VPN headend device in environments that have multiple (redundant) VPN headend devices.
- Ensures predictable failover time of remote sessions between headend devices when using IKE keepalives, especially in environments in which remote device route flapping is common (not taking into consideration the effects of route convergence, which may vary depending on the routing protocol used and the size of the network).
- Eliminates the need for the administration of static routes on upstream devices, as routes are dynamically learned by these devices.

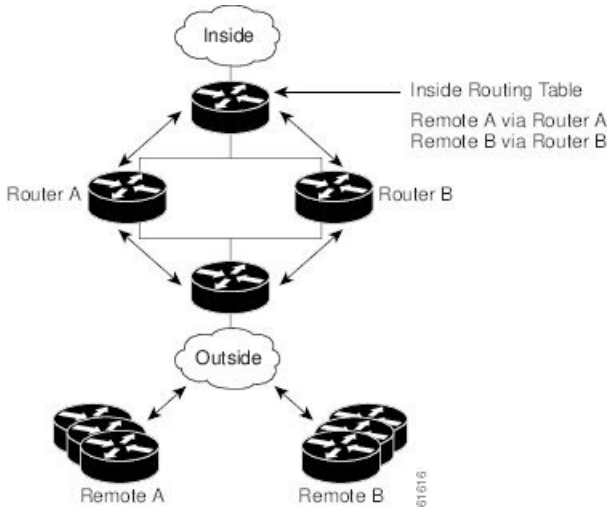
In the dynamic case, as remote peers establish IPsec security associations (SAs) with an RRI-enabled router, a static route is created for each subnet or host protected by that remote peer. For static crypto maps, a static route is created for each destination of an extended access list rule. When RRI is used on a static crypto map with an access control list (ACL), routes will always exist, even without the negotiation of IPsec SAs.

Note The use of any keyword in ACLs with RRI is not supported.

When routes are created, they are injected into any dynamic routing protocol and distributed to surrounding devices. This traffic flows, requiring IPsec to be directed to the appropriate RRI router for transport across the correct SAs to avoid IPsec policy mismatches and possible packet loss.

The figure below shows an RRI configuration functionality topology. Remote A is being serviced by Router A and Remote B connected to Router B, providing load balancing across VPN gateways at the central site. RRI on the central site devices ensures that the other router on the inside of the network can automatically make the correct forwarding decision. RRI also eliminates the need to administer static routes on the inside router.

Figure 97: Topology Showing Reverse Route Injection Configuration Functionality



Hot Standby Router Protocol and IPsec

Hot Standby Router Protocol (HSRP) provides high network availability by routing IP traffic from hosts on Ethernet networks without relying on the availability of any single router. HSRP is particularly useful for hosts that do not support a router discovery protocol, such as ICMP Router Discovery Protocol (IRDP) and do not have the functionality to switch to a new router when their selected router reloads or loses power. Without this functionality, a router that loses its default gateway because of a router failure cannot communicate with the network.

HSRP is configurable on LAN interfaces using standby command-line interface (CLI) commands. You can to use the standby IP address from an interface as the local IPsec identity or local tunnel endpoint.

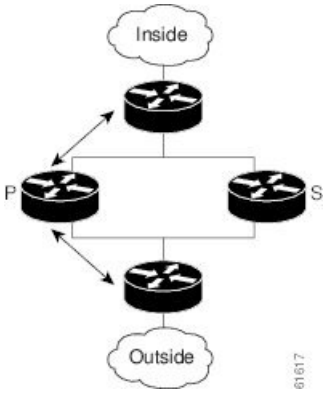
By using the standby IP address as the tunnel endpoint, failover can be applied to VPN routers by using HSRP. Remote VPN gateways connect to the local VPN router via the standby address that belongs to the active

device in the HSRP group. In the event of failover, the standby device takes over ownership of the standby IP address and begins to service remote VPN gateways.

Failover can be applied to VPN routers through the use of HSRP. Remote VPN gateways connect to the local VPN router through the standby address that belongs to the active device in the HSRP group. This functionality reduces configuration complexity on remote peers with respect to defining gateway lists, because only the HSRP standby address needs to be defined.

The figure below shows the enhanced HSRP functionality topology. Traffic is serviced by the active Router P, which is the active device in the standby group. In the event of failover, traffic is diverted to Router S, which is the original standby device. Router S assumes the role of the new active router and takes ownership of the standby IP address.

Figure 98: Topology Showing Hot Standby Router Protocol Functionality



Note In case of a failover, HSRP does not facilitate IPsec state information transference between VPN routers. This means that without this state transference, SAs to remotes will be deleted, requiring Internet Key Exchange (IKE) and IPsec SAs to be reestablished. To make IPsec failover more efficient, it is recommended that IKE keepalives be enabled on all routers.

How to Configure IPsec VPN High Availability Enhancements

Configuring Reverse Route Injection on a Dynamic Crypto Map

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same dynamic map name, but each with a different dynamic sequence number. Each member of the set may be configured for RRI.

To create a dynamic crypto map entry and enable RRI, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *map-name seq-num*

4. **set transform-set**
5. **reverse-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *map-name seq-num* **Example:** `Router(config)# crypto dynamic-map mymap`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**set transform-set** **Example:** `Router(config-crypto-m)#set transform-set`	Specifies which transform sets are allowed for the crypto map entry. Lists multiple transform sets in order of priority (highest priority first). This entry is the only configuration statement required in dynamic crypto map entries.
Step 5	**reverse-route** **Example:** `Router(config-crypto-m)#reverse-route`	Creates source proxy information.

Configuring Reverse Route Injection on a Static Crypto Map

Before configuring RRI on a static crypto map, note that:

- Routes are not created based on access list 102, as reverse-route is not enabled on mymap 2. RRI is not enabled by default and is not displayed in the router configuration.
- Enable a routing protocol to distribute the VPN routes to upstream devices.
- If Cisco Express Forwarding (CEF) is run on a VPN router configured for RRI, adjacencies need to be formed for each RRI injected network through the next hop device. As the next hop is not explicitly defined in the routing table for these routes, proxy-ARP should be enabled on the next hop router, which allows the CEF adjacency to be formed using the Layer 2 addresses of that device. In cases where there are many RRI injected routes, adjacency tables may become quite large, as an entry is created for each device from each of the subnets represented by the RRI route.

To add RRI to a static crypto map set, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set peer** *ip-address*
5. **reverse-route**
6. **match address**
7. **set transform-set** *transform-set-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)#crypto map mymap 3 ipsec-isakmp`	Adds a dynamic crypto map set to a static crypto map set and enters interface configuration mode.
Step 4	**set peer** *ip-address* **Example:** `Router(config-if)#set peer 209.165.200.248`	Specifies an IPsec peer IP address in a crypto map entry.
Step 5	**reverse-route** **Example:** `Router (config-if)#reverse-route`	Creates dynamic static routes based on crypto access control lists (ACLs).
Step 6	**match address** **Example:** `Router(config-if)# match address`	Specifies an extended access list for a crypto map entry.
Step 7	**set transform-set** *transform-set-name* **Example:** `Router (config-if)# set transform-set my_t_set1`	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first).

Configuring HSRP with IPsec

When configuring HSRP with IPsec, the following conditions may apply:

- When HSRP is applied to a crypto map on an interface, the crypto map must be reapplied if the standby IP address or the standby name is changed on that interface.
- If HSRP is applied to a crypto map on an interface, and you delete the standby IP address or the standby name from that interface, the crypto tunnel endpoint is reinitialized to the actual IP address of that interface.
- If you add the standby IP address and the standby name to an interface with the requirement IPsec failover, the crypto map must be reapplied with the appropriate redundancy information.
- Standby priorities should be equal on active and standby routers. If they are not, the higher priority router takes over as the active router. If the old active router comes back up and immediately assumes the active role before having time to report itself, standby and sync connections will be dropped.
- The IP addresses on the HSRP-tracked interfaces on the standby and active routers should both be either lower or higher on one router than the other. In the case of equal priorities (an HA requirement), HSRP will assign the active state-based IP address. If an addressing scheme exists so that the public IP address of router A is lower than the public IP address of router B, but the opposite is true for their private interfaces, an active/standby-standby/active split condition could exist, which will break connectivity.

Note To configure HSRP without IPsec, refer to the "Configuring IP Services" module in the *IP Application Services Configuration Guide*.

To apply a crypto map set to an interface, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **standby name** *group-name*
5. **standby ip** *ip-address*
6. **crypto map** *map-name* **redundancy** [*standby-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router#configure terminal`	
Step 3	**interface** *type slot / port* **Example:** `Router(config)#interface GigabitEthernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**standby name** *group-name* **Example:** `Router(config-if)#standby name mygroup`	Specifies the standby group name.
Step 5	**standby ip** *ip-address* **Example:** `Router(config-if)#standby ip 209.165.200.249`	Specifies the IP address of the standby groups • This command is required for one device in the group.
Step 6	**crypto map** *map-name* **redundancy** [*standby-name*] **Example:** `Router (config-if)#crypto map mymap redundancy`	Specifies the IP redundancy address as the tunnel endpoint for IPsec.

Verifying VPN IPsec Crypto Configuration

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec transform-set**
3. **show crypto map** [**interface** *interface* | **tag** *map-name*]
4. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**]
5. **show crypto dynamic-map** [**tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec transform-set** **Example:** `Router#show crypto ipsec transform-set`	Displays the transform set configuration.

	Command or Action	Purpose
Step 3	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router#show crypto map tag mycryptomap`	Displays your crypto map configuration.
Step 4	**show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**] **Example:** `Router#show crypto ipsec sa address detail`	Displays information about IPsec SAs.
Step 5	**show crypto dynamic-map** [**tag** *map-name*] **Example:** `Router#show crypto dynamic-map tag mymap`	Displays information about dynamic crypto maps.

Configuration Examples for IPsec VPN High Availability Enhancements

Example: Configuring Reverse Route Injection on a Dynamic Crypto Map

In the following example, using the **reverse-route** command in the definition of the dynamic crypto map template ensures that routes are created for any remote proxies (subnets or hosts), protected by the connecting remote IPsec peers.

```
crypto dynamic mydynmap 1
 set transform-set my-transform-set
 reverse-route
```

This template is then associated with a “parent” crypto map statement and then applied to an interface.

```
crypto map mymap 3 ipsec-isakmp dynamic mydynmap
 interface FastEthernet 0/0
 crypto map mymap
```

Example: Configuring Reverse Route Injection on a Static Crypto Map

RRI is a good solution for topologies that require encrypted traffic to be diverted to a VPN router and all other traffic to a different router. In these scenarios, RRI eliminates the need to manually define static routes on devices.

RRI is not required if a single VPN router is used, and all traffic passes through the VPN router during its path in to and out of the network.

If you choose to manually define static routes on the VPN router for remote proxies and have these routes permanently installed in the routing table, RRI should not be enabled on the crypto map instance that covers

the same remote proxies. In this case, there is no possibility of user-defined static routes being removed by RRI.

Routing convergence can affect the success of a failover based on the routing protocol used to advertise routes (link state versus periodic update). We recommend that a link state routing protocol such as OSPF be used to help speed convergence time by ensuring that routing updates are sent as soon as a change in routing state is detected.

In the following example, RRI is enabled for mymap 1, but not for mymap 2. Upon the application of the crypto map to the interface, a route is created based on access-list 101 analogous to the following:

```
IP route 172.17.11.0 255.255.255.0 FastEthernet 0/0
crypto map mymap 1 ipsec-isakmp
 set peer 172.17.11.1
 reverse-route
 set transform-set my-transform-set
 match address 101
crypto map mymap 2 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set my-transform-set
 match address 102
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
interface FastEthernet 0/0
 crypto map mymap
```

Example: Configuring HSRP with IPsec

The following example shows how all remote VPN gateways connect to the router via 192.168.0.3. The crypto map on the interface binds this standby address as the local tunnel endpoint for all instances of the crypto map named *mymap* and at the same time ensures that HSRP failover is facilitated between an active and standby device belonging to the same standby group named group1.

Note that RRI also provides the ability for only the active device in the HSRP group to be advertising itself to inside devices as the next hop VPN gateway to the remote proxies. If there is a failover, routes are deleted on the formerly active device and created on the newly active device.

```
crypto map mymap 1 ipsec-isakmp
 set peer 10.1.1.1
 reverse-route
 set transform-set esp-aes-sha
 match address 102
Interface FastEthernet 0/0
 ip address 192.168.0.2 255.255.255.0
 standby name group1
 standby ip 192.168.0.3
 crypto map mymap redundancy group1
access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

The standby name needs to be configured on all devices in the standby group, and the standby address needs to configured on at least one member of the group. If the standby name is removed from the router, the IPsec SAs will be deleted. If the standby name is added again, regardless of whether the same name or a different name is used, the crypto map (using the redundancy option) will have to be reapplied to the interface.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring HSRP without IPsec	"Configuring IP Services" module in the *IP Application Services Configuration Guide*
Configuring stateful failover for IP security (IPsec)	"Stateful Failover for IPsec" module in the *Security Configuration Guide: Secure Connectivity*
Recommended cryptographic algorithms	Next Generation Encryption

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec VPN High Availability Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 261: Feature Information for IPsec VPN High Availability Enhancements

Feature Name	Releases	Feature Information
IPsec VPN High Availability Enhancements	Cisco IOS XE 3.1.0S	The IPsec VPN High Availability Enhancements feature consists of two features:Reverse Route Injection (RRI) and Hot Standby Router Protocol (HSRP) with IPsec. When used together, these two features provide you with a simplified network design for VPNs and reduced configuration complexity on remote peers when defining gateway lists. The following commands were introduced or modified: **crypto map** (interface IPsec), **reverse-route**.

CHAPTER 194

IPsec Preferred Peer

The IP Security (IPsec) Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario.

This feature includes the following capabilities:

- Default peer configuration
- IPsec idle-timer usage with default peer

Prerequisites for IPsec Preferred Peer

- You must have a properly defined, complete crypto map.

Restrictions for IPsec Preferred Peer

Default Peer

- This feature must be used in conjunction with Dead Peer Detection (DPD). It is most effective on a remote site running DPD in periodic mode. DPD detects the failure of a device quickly and resets the peer list so that the default peer is tried for the next attempted connection.
- Only one peer can be designated as the default peer in a crypto map.
- The default peer must be the first peer in the peer list.

IPsec Idle Timer Usage with Default Peer

- This feature works only on the crypto map for which it is configured. You cannot configure the capability globally for all crypto maps.
- If there is a global idle timer, the crypto map idle-timer value must be different from the global value; otherwise, the idle timer is not added to the crypto map.

IPsec Failover

IPsec on the Cisco ASR 1000 Series Router supports only stateless failover. IPsec failover is a feature that increases the total uptime (or availability) of an IPsec network. This is accomplished traditionally by employing a redundant (standby) router in addition to the original (active) router. If the active router becomes unavailable for any reason, the standby router takes over the processing of IKE and IPsec.

IPsec failover falls into two categories: stateless failover and stateful failover. Stateless failover uses protocols such as the Hot Standby Router Protocol (HSRP) to provide primary-to-secondary cutover and also allows the active and standby VPN gateways to share a common virtual IP address.

Information About IPsec Preferred Peer

IPsec

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating Internet Protocol (IP) packets between participating IPsec devices (peers), such as Cisco routers.

IPsec provides the following network security services. These services are optional. In general, local security policy dictates the use of one or more of these services:

- Data Confidentiality--The IPsec sender can encrypt packets before transmitting them across a network.
- Data Integrity--The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication--The IPsec receiver can authenticate the source of the IPsec packets sent.
- Anti-Replay--The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

IPsec provides secure tunnels between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying characteristics of these tunnels. When the IPsec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

Dead Peer Detection

The VPN Client uses a keepalive mechanism called Dead Peer Detection (DPD) to check the availability of the VPN device on the other side of an IPsec tunnel. If the network is unusually busy or unreliable, you can increase the number of seconds that the VPN Client will wait before deciding whether the peer is no longer active.

Keepalive packets are not sent if traffic is received. This lowers the overhead associated with DPD, because on a heavily loaded network very few keepalive packets will be sent because traffic is being received on the tunnels. In addition, DPD sends keepalive packets only if there is user traffic to send (and no user traffic is received).

You can configure Internet Key Exchange (IKE) DPD so that DPD sends the keepalive packets whether or not there is outbound user data. That is, as long as there is no inbound user data, the keepalive packets are sent at the configured keepalive interval.

Default Peer Configuration

If a connection timeout occurs, the connection to the current peer is closed. The **set peer** command allows you to configure the first peer as the default peer. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If the default peer is unresponsive, the next peer in the peer list becomes the current peer and future connections through the crypto map try that peer.

This capability is useful when traffic on a physical link stops due to the failure of a remote peer. DPD indicates that the remote peer is unavailable, but that peer remains the current peer.

A default peer facilitates the failover to a preferred peer that was previously unavailable, but has returned to service. Users can give preference to certain peers in the event of a failover. This is useful if the original failure was due to a network connectivity problem rather than failure of the remote peer.

Idle Timers

When a router creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers.

IPsec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPsec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required.

If IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

IPsec Idle-Timer Usage with Default Peer

If all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer configured in the **set peer** command. If a default peer is not configured and there is a connection timeout, the current peer remains the one that timed out.

This enhancement helps facilitate a failover to a preferred peer that was previously unavailable but is in service now.

Peers on Crypto Maps

A crypto map set can contain multiple entries, each with a different access list. The router searches the crypto map entries in order, and attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as Cisco, connections are established with the remote peer as specified in the set peer statements within the crypto map.

How to Configure IPsec Preferred Peer

Configuring a Default Peer

To configure a default peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]
4. **set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**] }
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set peer** {*host-name* [**dynamic**] [**default**]	*ip-address* [**default**] } **Example:**	Specifies an IPsec peer in a crypto map entry. Ensures that the first peer specified is defined as the default peer.

	Command or Action	Purpose
	`Router(config-crypto-map)# set peer 10.0.0.2 default`	
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuring the Idle Timer

To configure the idle timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]
4. **set security-association idletime** *seconds* [**default**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set security-association idletime** *seconds* [**default**] **Example:** `Router(config-crypto-map)# set security-association idletime 120 default`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuration Examples for IPsec Preferred Peer

Configuring a Default Peer Example

The following example shows that the first peer, at IP address 10.1.1.1, is the default peer:

```
crypto map tohub 1 ipsec-isakmp
 set peer 10.1.1.1 default
 set peer 10.2.2.2
```

Configuring the IPsec Idle Timer Example

In the following example, if the current peer is idle for 120 seconds, the default peer 10.1.1.1 (which was specified in the **set peer**command) is used for the next attempted connection:

```
crypto map tohub 1 ipsec-isakmp
 set peer 10.1.1.1 default
 set peer 10.2.2.2
 set security-association idletime 120 default
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPsec	*Security for VPNs with IPsec*
Crypto map	• *Security for VPNs with IPsec* • *Configuring Internet Key Exchange for IPsec VPNs*
DPD	*IPsec Dead Peer Detection Periodic Message Option*
Security commands	*Cisco IOS Security Command Reference*

MIBs

MIBs	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec Preferred Peer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 262: Feature Information for IPsec Preferred Peer

Feature Name	Releases	Feature Information
IPsec Preferred Peer	Cisco IOS XE Release 2.1	The IPsec Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario. The following commands were introduced or modified: **set peer** (IPsec) and **set security-association idle-time**.

Glossary

crypto access list --A list that defines which IP traffic will be protected by crypto and which traffic will not be protected by crypto.

crypto map --A map that specifies which traffic should be protected by IPsec, where IPsec-protected traffic should be sent, and what IPsec transform sets should be applied to this traffic.

dead peer detection --A feature that allows the router to detect an unresponsive peer.

keepalive message --A message sent by one network device to inform another network device that the virtual circuit between the two is still active.

peer --Router or other device that participates in IPsec and IKE. In IPsec, peers are devices or entities that communicate securely either through the exchange of keys or the exchange of digital certificates.

SA --security association. An instance of security policy and keying material applied to a data flow. Both IKE and IPsec use SAs, although SAs are independent of one another. IPsec SAs are unidirectional and are unique in each security protocol. An IKE SA is used by IKE only, and unlike the IPsec SA, it is bidirectional. IKE negotiates and establishes SAs on behalf of IPsec. A user also can establish IPsec SAs manually. A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports Encapsulating Security Payload (ESP) between peers, one ESP SA is required for each direction. SAs are identified uniquely by destination (IPsec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

transform set --An acceptable combination of security protocols, algorithms, and other settings to apply to IPsec-protected traffic. During the IPsec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

CHAPTER 195

Real-Time Resolution for IPsec Tunnel Peer

After a user specifies a host name (instead of an IP address) for remote IP Security (IPsec) peer, the Real-Time Resolution for IPsec Tunnel Peer feature allows the host name to be domain name server (DNS) resolved before the router establishes the IPsec tunnel. Thus, the router can immediately discover whether the IP address of the peer has changed.

Restrictions for Real-Time Resolution for IPsec Tunnel Peer

Secure DNS Requirement

It is recommended that you use this feature only with secure DNS and when the DNS responses can be authenticated. Otherwise, an attacker can spoof or forge DNS responses and have access to Internet Key Exchange (IKE) authentication data, such as a certificate. If an attacker has a certificate that is trusted by the initiating host, the attacker can successfully establish Phase 1 IKE security association (SA), or the attacker can try to guess the preshared key that is shared between the initiator and the actual responder.

DNS Initiator

DNS names resolution for remote IPsec peers will work only if they are used as an initiator. The first packet that is to be encrypted will trigger a DNS lookup; after the DNS lookup is complete, subsequent packets will trigger IKE.

Information About Real-Time Resolution for IPsec Tunnel Peer

Real-Time Resolution Via Secure DNS

When specifying the host name of a remote IPsec peer via the **set peer** command, you can also issue the **dynamic** keyword, which defers DNS resolution of the host name until right before the IPsec tunnel has been

established. Deferring resolution enables the software to detect whether the IP address of the remote IPsec peer has changed. Thus, the software can contact the peer at the new IP address.

If the **dynamic** keyword is not issued, the host name is resolved immediately after it is specified. So, the software cannot detect an IP address change and, therefore, attempts to connect to the IP address that it previously resolved.

DNS resolution assures users that their established IPsec tunnel is secure and authenticated.

How to Configure Real-Time Resolution

Configuring Real-Time Resolution for IPsec Peers

Use this task to configure a router to perform real-time DNS resolution with a remote IPsec peer; that is, the host name of peer is resolved via a DNS lookup right before the router establishes a connection (an IPsec tunnel) with the peer.

Before you begin

Before creating a crypto map, you should perform the following tasks:

- Define Internet Security Association Key Management Protocol (ISAKMP) policies.
- Define IPsec transform sets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **match address** *access-list-id*
5. **set peer** {*host-name* [**dynamic**] | *ip-address*
6. **set transform-set** *transform-set-name1* [*transform-set-name2 ... transform-set-name6*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map secure_b 10 ipsec-isakmp`	Specifies the crypto map entry to create (or modify) and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 140`	Names an extended access list. This access list determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry.

Step 5	**set peer** {*host-name* [**dynamic**] | *ip-address*
	Example:
	`Router(config-crypto-m)#`
	`set peer b.cisco.com dynamic`
		Specifies a remote IPsec peer.
		This is the peer to which IPsec-protected traffic can be forwarded.
		• **dynamic** --Allows the host name to be resolved via a DNS lookup just before the router establishes the IPsec tunnel with the remote peer. If this keyword is not specified, the host name will be resolved immediately after the host name is specified.
		Repeat for multiple remote peers.

Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2 ... transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set myset`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Troubleshooting Tips

To display crypto map configuration information, use the **show crypto map** command.

What to Do Next

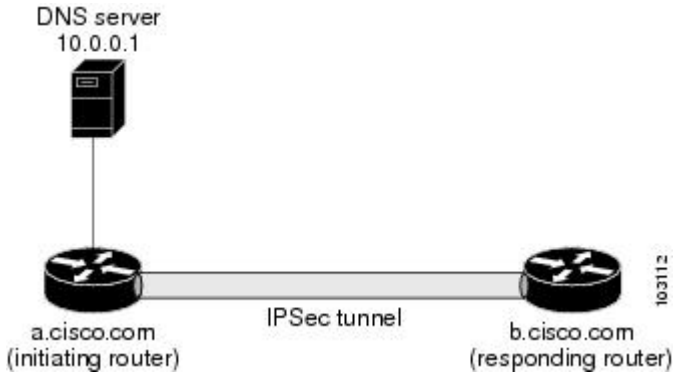
You need to apply a crypto map set to each interface through which IPsec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association (SA) negotiation on behalf of traffic to be protected by crypto.

Configuration Examples for Real-Time Resolution

Configuring Real-Time Resolution for an IPsec Peer Example

The figure below and the following example illustrate how to create a crypto map that configures the host name of a remote IPsec peer to DNS resolved via a DNS lookup right before the software attempts to establish a connection with that peer.

Figure 99: Real-Time Resolution Sample Topology




```
! Configure the initiating router.
hostname a.cisco.com
ip domain name cisco.com
ip name server 10.0.0.1
!
crypto map secure_b 10 ipsec-isakmp
  match address 140
  set peer b.cisco.com dynamic
  set transform-set xset
interface serial1
  ip address 10.10.0.1
  crypto map secure_b
access-list 140 permit ...
!
! Configure the responding router (the remote IPSec peer).
hostname b.cisco.com
!
crypto map secure_a 10 ipsec-isakmp
  match address 150
  set peer 10.10.0.1
  set transform-set
interface serial0/1
  ip address 10.0.0.1
  crypto map secure_a
access-list 150 ...
! DNS server configuration
b.cisco.com 10.0.0.1       # the address of serial0/1 of b.cisco.com
```

Additional References

Related Documents

Related Topic	Document Title
Crypto maps	"Configuring Security for VPNs with IPsec" module in the *Security for VPNs with IPsec Configuration Guide*
ISAKMP policies	"Configuring Internet Key Exchange for IPsec VPNs" module in the *Internet Key Exchange for IPsec VPNs Configuration Guide*
IPsec and IKE configuration commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Real-Time Resolution for IPsec Tunnel Peer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 263: Feature Information for Real-Time Resolution for IPsec Tunnel Peer

Feature Name	Releases	Feature Information
Real-Time Resolution for IPsec Tunnel Peer	Cisco IOS XE Release 2.1	After a user specifies a host name (instead of an IP address) for remote IP Security (IPsec) peer, this feature allows the host name to be domain name server (DNS) resolved before the router establishes the IPsec tunnel. Thus, the router can immediately discover whether the IP address of the peer has changed. The following commands were introduced or modified: **set peer (IPsec)**.

PART XX

Internet Key Exchange

CHAPTER 196

Configuring Internet Key Exchange for IPsec VPNs

This module describes how to configure the Internet Key Exchange (IKE) protocol for basic IP Security (IPsec) Virtual Private Networks (VPNs). IKE is a key management protocol standard that is used in conjunction with the IPsec standard. IPsec is an IP security feature that provides robust authentication and encryption of IP packets.

IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

IKE is a hybrid protocol, that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for IKE Configuration

- You should be familiar with the concepts and tasks explained in the module Configuring Security for VPNs with IPsec.
- Ensure that your Access Control Lists (ACLs) are compatible with IKE. Because IKE negotiation uses User Datagram Protocol (UDP) on port 500, your ACLs must be configured so that UDP port 500 traffic

is not blocked at interfaces used by IKE and IPsec. In some cases you might need to add a statement to your ACLs to explicitly permit UDP port 500 traffic.

Restrictions for IKE Configuration

- To avoid profiles being locked or leading to DMI degrade state, before using the **config-replace** command to replace a configuration, ensure to shut down the tunnel interface to bring down all crypto sessions, and tunnel configurations.
- The initiating router *must not* have a certificate associated with the remote peer.
- The preshared key *must* be by a fully qualified domain name (FQDN) on both peers. (To configure the preshared key, enter the **crypto isakmp key** command.)
- The communicating routers *must* have a FQDN host entry for each other in their configurations.
- The communicating routers *must* be configured to authenticate by hostname, *not* by IP address; thus, you should use the **crypto isakmp identity hostname** command.
- Use **show crypto eli** command to determine the software encryption limitations for your device. Without any hardware modules, the limitations are as follows:
  - 1000 IPsec security associations (SAs)
  - 100 IKE SAs
  - 50 Diffie-Hellman (DH) session keys
- Disable the crypto batch functionality, by using the **no crypto batch allowed** command to increase the performance of a TCP flow on a Site-to-site VPN. However, disabling the crypto batch functionality might have an impact on CPU utilization.
- Starting with Cisco IOS Release 15.0(1)SY and later, you cannot configure IPSec Network Security features using **crypto ipsec** commands on Cisco Catalyst 6500 Series switches. For IPSec support on these switches, you must use a hardware encryption engine.

Information About Configuring IKE for IPsec VPNs

Supported Standards for Use with IKE

Cisco implements the following standards:

- IPsec—IP Security Protocol. IPsec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

- ISAKMP—Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.
- Oakley—A key exchange protocol that defines how to derive authenticated keying material.
- Skeme—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

Note Cisco no longer recommends using DES, 3DES, MD5 (including HMAC variant), and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use AES,SHA-256 and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The component technologies implemented for use by IKE include the following:

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is privacy transform for IPsec and IKE and has been developed to replace the Data Encryption Standard (DES). AES is designed to be more secure than DES: AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet.

Cisco IOS software also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers, particularly in the finance industry, to utilize network-layer encryption.

Note Cisco IOS images that have strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images that are to be installed outside the United States require an export license. Customer orders might be denied or subject to delay because of United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- SEAL—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact to the CPU when compared to other software-based algorithms.
- SHA-2 and SHA-1 family (HMAC variant)—Secure Hash Algorithm (SHA) 1 and 2. Both SHA-1 and SHA-2 are hash algorithms used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. SHA-2 family adds the SHA-256 bit hash algorithm and SHA-384 bit hash algorithm. This functionality is part of the Suite-B requirements that comprises four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the

Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.

- RSA signatures and RSA encrypted nonces—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provide nonrepudiation, and RSA encrypted nonces provide repudiation. (Repudation and nonrepudation have to do with traceability.)
- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. It supports 768-bit (the default), 1024-bit, 1536-bit, 2048-bit, 3072-bit, and 4096-bit DH groups. It also supports a 2048-bit DH group with a 256-bit subgroup, and 256-bit and 384-bit elliptic curve DH (ECDH). Cisco recommends using 2048-bit or larger DH key exchange, or ECDH key exchange.
- MD5—Message Digest 5 (Hash-Based Message Authentication Code (HMAC) variant). A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.

IKE interoperates with the X.509v3 certificates, which are used with the IKE protocol when authentication requires public keys. This certificate support allows the protected network to scale by providing the equivalent of a digital ID card to each device. When two devices intend to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer).

IKE Benefits

IKE automatically negotiates IPsec security associations (SAs) and enables IPsec secure communications without costly manual preconfiguration. Specifically, IKE provides the following benefits:

- Allows you to specify a lifetime for the IPsec SA.
- Allows encryption keys to change during IPsec sessions.
- Allows IPsec to provide antireplay services.
- Permits certification authority (CA) support for a manageable, scalable IPsec implementation.
- Allows dynamic authentication of peers.

IKE Main Mode and Aggressive Mode

IKE has two phases of key negotiation: phase 1 and phase 2. Phase 1 negotiates a security association (a key) between two IKE peers. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During phase 2 negotiation, IKE establishes keys (security associations) for other applications, such as IPsec.

Phase 1 negotiation can occur using main mode or aggressive mode. Main mode tries to protect all information during the negotiation, meaning that no information is available to a potential attacker. When main mode is used, the identities of the two IKE peers are hidden. Although this mode of operation is very secure, it is relatively costly in terms of the time required to complete the negotiation. Aggressive mode takes less time to negotiate keys between peers; however, it gives up some of the security provided by main mode negotiation. For example, the identities of the two parties trying to establish a security association are exposed to an eavesdropper.

The two modes serve different purposes and have different strengths. Main mode is slower than aggressive mode, but main mode is more secure and more flexible because it can offer an IKE peer more security proposals than aggressive mode. Aggressive mode is less flexible and not as secure, but much faster.

In Cisco IOS software, the two modes are not configurable. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode; however, in cases where there is no corresponding information to initiate authentication, and there is a preshared key associated with the hostname of the peer, Cisco IOS software can initiate aggressive mode. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

IKE Policies Security Parameters for IKE Negotiation

An IKE policy defines a combination of security parameters to be used during the IKE negotiation. You must create an IKE policy at each peer participating in the IKE exchange.

If you do not configure any IKE policies, your router will use the default policy, which is always set to the lowest priority and which contains the default value of each parameter.

About IKE Policies

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer--e ach with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Tip If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

IKE Peers Agreeing Upon a Matching IKE Policy

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values.

If a match is found, IKE will complete negotiation, and IPsec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPsec will not be established.

Note The parameter values apply to the IKE negotiations after the IKE SA is established. Depending on the authentication method specified in a policy, additional configuration might be required. For more information, see Configuring IKE Authentication, on page 2539.

If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

IKE Authentication

IKE authentication consists of the following options and each authentication method requires additional configuration.

RSA Signatures

With RSA signatures, you can configure the peers to obtain certificates from a CA. (The CA must be properly configured to issue the certificates.) Using a CA can dramatically improve the manageability and scalability of your IPsec network. Additionally, RSA signature-based authentication uses only two public key operations, whereas RSA encryption uses four public key operations, making it costlier in terms of overall performance. To properly configure CA support, see the module "Deploying RSA Keys Within a PKI.

The certificates are used by each peer to exchange public keys securely. RSA signatures require that each peer has the public signature key of the remote peer. When both peers have valid certificates, they will automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

You can also exchange the public keys manually, as described in Configuring RSA Keys Manually for RSA Encrypted Nonces, on page 2539.

RSA signatures provide nonrepudiation for the IKE negotiation. And, you can prove to a third party after the fact that you did indeed have an IKE negotiation with the remote peer.

RSA Encrypted Nonces

With RSA encrypted nonces, you must ensure that each peer has the public keys of the other peers.

Unlike RSA signatures, the RSA encrypted nonces method cannot use certificates to exchange public keys. Instead, ensure that each peer has the other's public keys by one of the following methods:

- Configuring RSA keys manually as described in Configuring RSA Keys Manually for RSA Encrypted Nonces, on page 2539.
- Ensuring that an IKE exchange using RSA signatures with certificates has already occurred between the peers. (The peers' public keys are exchanged during the RSA-signatures-based IKE negotiations if certificates are used.) To make that the IKE exchange happens, specify two policies: a higher-priority policy with RSA encrypted nonces and a lower-priority policy with RSA signatures. When IKE negotiations occur, RSA signatures will be used the first time because the peers do not yet have each other's public keys. Then future IKE negotiations can use RSA encrypted nonces because the public keys will have been exchanged. This alternative requires that you already have CA support configured.

RSA encrypted nonces provide repudiation for the IKE negotiation; however, unlike RSA signatures, you cannot prove to a third party that you had an IKE negotiation with the remote peer.

Preshared Keys

Preshared Keys An Overview

Preshared keys are clumsy to use if your secured network is large, and they do not scale well with a growing network. However, they do not require use of a CA, as do RSA signatures, and might be easier to set up in a small network with fewer than ten nodes. RSA signatures also can be considered more secure when compared with preshared key authentication.

Note If RSA encryption is configured and signature mode is negotiated (and certificates are used for signature mode), the peer will request both signature and encryption keys. Basically, the router will request as many keys as the configuration will support. If RSA encryption is not configured, it will just request a signature key.

ISAKMP Identity Setting for Preshared Keys

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPsec SAs, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IP address, depending on how you have set the ISAKMP identity of the router.

By default, a peer’s ISAKMP identity is the IP address of the peer. If appropriate, you could change the identity to be the peer's hostname instead. As a general rule, set the identities of all peers the same way--either all peers should use their IP addresses or all peers should use their hostnames. If some peers use their hostnames and some peers use their IP addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a Domain Name System (DNS) lookup is unable to resolve the identity.

Mask Preshared Keys

A mask preshared key allows a group of remote users with the same level of authentication to share an IKE preshared key. The preshared key of the remote peer must match the preshared key of the local peer for IKE authentication to occur.

A m ask preshared key is usually distributed through a secure out-of-band channel. In a remote peer-to-local peer scenario, any remote peer with the IKE preshared key configured can establish IKE SAs with the local peer.

If you specify the **mask** keyword with the **crypto isakmp key** command, it is up to you to use a subnet address, which will allow more peers to share the same key. That is, the preshared key is no longer restricted to use between two users.

Note Using 0.0.0.0 as a subnet address is not recommended because it encourages group preshared keys, which allow all peers to have the same group key, thereby reducing the security of your user authentication.

Disable Xauth on a Specific IPsec Peer

Disabling Extended Authentication (Xauth) for static IPsec peers prevents the routers from being prompted for Xauth information--username and password.

IKE Mode Configuration

IKE mode configuration, as defined by the Internet Engineering Task Force (IETF), allows a gateway to download an IP address (and other network-level configuration) to the client as part of an IKE negotiation. Using this exchange, the gateway gives an IP address to the IKE client to be used as an "inner" IP address encapsulated under IPsec. This method provides a known IP address for the client that can be matched against IPsec policy.

To implement IPsec VPNs between remote access clients that have dynamic IP addresses and a corporate gateway, you have to dynamically administer scalable IPsec policy on the gateway once each client is authenticated. With IKE mode configuration, the gateway can set up a scalable policy for a very large set of clients regardless of the IP addresses of those clients.

There are two types of IKE mode configuration:

- Gateway initiation--Gateway initiates the configuration mode with the client. Once the client responds, the IKE modifies the identity of the sender, the message is processed, and the client receives a response.
- Client initiation--Client initiates the configuration mode with the gateway. The gateway responds with an IP address that it has allocated for the client.

How to Configure IKE for IPsec VPNs

If you do not want IKE to be used with your IPsec implementation, you can disable it at all IPsec peers via the **no crypto isakmp** command, skip the rest of this chapter, and begin your IPsec VPN.

IKE is enabled by default. IKE does not have to be enabled for individual interfaces, but it is enabled globally for all interfaces at the router.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Perform the following tasks to provide authentication of IPsec peers, negotiate IPsec SAs, and establish IPsec keys:

Troubleshooting Tips

- Clear (and reinitialize) IPsec SAs by using the **clear crypto sa** EXEC command.

Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command in the Cisco IOS Security Command Reference.

- The default policy and default values for configured policies do not show up in the configuration when you issue the **show running-config** command. To display the default policy and any default values within configured policies, use the **show crypto isakmp policy** command.

- Any IPsec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

If a user enters an IPsec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPsec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

What to Do Next

Depending on which authentication method you specified in your IKE policies (RSA signatures, RSA encrypted nonces, or preshared keys), you must do certain additional configuration tasks before IKE and IPsec can successfully use the IKE policies. For information on completing these additional tasks, see Configuring IKE Authentication, on page 2539.

To configure an AES-based transform set, see the module "Configuring Security for VPNs with IPsec."

Configuring IKE Authentication

After you have created at least one IKE policy in which you specified an authentication method (or accepted the default method), you need to configure an authentication method. IKE policies cannot be used by IPsec until the authentication method is successfully configured.

Note Before configuring IKE authentication, you must have configured at least one IKE policy, which is where the authentication method was specified (or RSA signatures was accepted by default).

To configure IKE authentication, you should perform one of the following tasks, as appropriate:

Prerequisites

You must have configured at least one IKE policy, which is where the authentication method was specified (or RSA signatures was accepted by default).

Configuring RSA Keys Manually for RSA Encrypted Nonces

Note This task can be performed only if a CA is not in use.

To manually configure RSA keys, perform this task for each IPsec peer that uses RSA encrypted nonces in an IKE policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa**{**general-keys**} | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*]

4. **crypto key generate ec keysize** [**256** | **384**] [**label** *label-string*]
5. **exit**
6. **show crypto key mypubkey rsa**
7. **configure terminal**
8. **crypto key pubkey-chain rsa**
9. Do one of the following:
   - **named-key** *key-name* [**encryption** | **signature**]
   - **addressed-key** *key-address* [**encryption** | **signature**]
10. **address** *ip-address*
11. **key-string** key-string
12. **quit**
13. Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.
14. **exit**
15. **exit**
16. **show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa**{**general-keys**} | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] **Example:** `Router(config)# crypto key generate rsa general-keys modulus 360`	Generates the RSA keys. • If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.
Step 4	**crypto key generate ec keysize** [**256** | **384**] [**label** *label-string*] **Example:** `Router(config)# crypto key generate ec keysize 256 label Router_1_Key`	Generates the EC keys. • The 256 keyword specifies a 256-bit keysize. • The 384 keyword specifies a 384-bit keysize. • A label can be specified for the EC key by using the **label** keyword and *label-string* argument. **Note** If a label is not specified, then FQDN value is used.
Step 5	**exit** **Example:**	(Optional) Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 6	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the generated RSA public keys.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Returns to global configuration mode.
Step 8	**crypto key pubkey-chain rsa** **Example:** `Router(config)# crypto key pubkey-chain rsa`	Enters public key chain configuration mode (so you can manually specify the RSA public keys of other devices).

Step 9	Do one of the following:
	• **named-key** *key-name* [**encryption** | **signature**]
	• **addressed-key** *key-address* [**encryption** | **signature**]
	Example:
	`Router(config-pubkey-chain)# named-key otherpeer.example.com`
	Example:
	`Router(config-pubkey-chain)# addressed-key 10.1.1.2 encryption`
		Indicates which remote peer's RSA public key you will specify and enters public key configuration mode.
		• If the remote peer uses its hostname as its ISAKMP identity, use the **named-key** command and specify the remote peer's FQDN, such as somerouter.example.com, as the *key-name*.
		• If the remote peer uses its IP address as its ISAKMP identity, use the **addressed-key** command and specify the remote peer's IP address as the *key-address*.

Step 10	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	Specifies the IP address of the remote peer. • If you use the **named-key** command, you need to use this command to specify the IP address of the peer.

Step 11	**key-string** key-string
	Example:
	`Router(config-pubkey-key)# key-string`
	Example:
	`Router(config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973`
	Example:
	`Router(config-pubkey)# 2DD50A37 C4F4B0FD 9DADE748 429618D5`
	Example:
	`Router(config-pubkey)# 18242BA3 2EDFBDD3 4296142A DDF7D3D8`
	Example:
	`Router(config-pubkey)# 08407685 2F2190A0 0B43F1BD 9A8A26DB`
	Example:
		Specifies the RSA public key of the remote peer.
		• (This key was previously viewed by the administrator of the remote peer when the RSA keys of the remote router were generated.)

	Command or Action	Purpose

	`Router(config-pubkey)# 07953829 791FCDE9 A98420F0 6A82045B`
	Example:
	`Router(config-pubkey)# 90288A26 DBC64468 7789F76E EE21`	

Step 12	**quit** **Example:** `Router(config-pubkey-key)# quit`	Returns to public key chain configuration mode.
Step 13	Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.	—
Step 14	**exit** **Example:** `Router(config-pubkey-key)# exit`	Returns to global configuration mode.
Step 15	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 16	**show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*] **Example:** `Router# show crypto key pubkey-chain rsa`	(Optional) Displays either a list of all RSA public keys that are stored on your router or details of a particular RSA key that is stored on your router.

Configuring Preshared Keys

To configure preshared keys, perform these steps for each peer that uses preshared keys in an IKE policy.

Note Preshared keys do not scale well with a growing network. Mask preshared keys have the following restrictions:

- The SA cannot be established between the IPsec peers until all IPsec peers are configured for the same preshared key.
- The mask preshared key must be distinctly different for remote users requiring varying levels of authorization. You must configure a new preshared key for each level of trust and assign the correct keys to the correct parties. Otherwise, an untrusted party may obtain access to protected data.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp identity** {**address** | **dn** | **hostname**}
4. **ip host** *hostname address1* [*address2...address8*]
5. Do one of the following:

- **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]
- **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**]

6. Do one of the following:
   - **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]
   - **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**]
7. Repeat these steps at each peer that uses preshared keys in an IKE policy.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp identity** {**address** | **dn** | **hostname**} **Example:** `Router(config)# crypto isakmp identity address`	Specifies the peer's ISAKMP identity by IP address, by distinguished name (DN) hostname at the local peer. • **address**--Typically used when only one interface (and therefore only one IP address) will be used by the peer for IKE negotiations, and the IP address is known. • **dn**--Typically used if the DN of a router certificate is to be specified and chosen as the ISAKMP identity during IKE processing. The dn keyword is used only for certificate-based authentication. • **hostname**--Should be used if more than one interface on the peer might be used for IKE negotiations, or if the interface's IP address is unknown (such as with dynamically assigned IP addresses).
Step 4	**ip host** *hostname address1* [*address2...address8*] **Example:** `Router(config)# ip host RemoteRouter.example.com 192.168.0.1`	If the local peer's ISAKMP identity was specified using a hostname, maps the peer's host name to its IP address(es) at all the remote peers. (This step might be unnecessary if the hostname or address is already mapped in a DNS server.)
Step 5	Do one of the following: • **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]	Specifies at the local peer the shared key to be used with a particular remote peer. • If the remote peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

	Command or Action	Purpose

	• **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**]
	Example:
	`Router(config)# crypto isakmp key sharedkeystring address 192.168.1.33 no-xauth`
	Example:
	`Router(config) crypto isakmp key sharedkeystring hostname RemoteRouter.example.com`
		• **no-xauth**--Prevents the router from prompting the peer for Xauth information.
		Note According to the design of preshared key authentication in IKE main mode, preshared keys must be based on the IP address of the peers. Although you can send a hostname as the identity of a preshared key authentication, the key is searched on the IP address of the peer; if the key is not found (based on the IP address) the negotiation will fail.

Step 6	Do one of the following:
	• **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]
	• **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**]
	Example:
	`Router(config) crypto isakmp key sharedkeystring address 10.0.0.1`
	Example:
	`Router(config) crypto isakmp key sharedkeystring hostname LocalRouter.example.com`
		Specifies at the remote peer the shared key to be used with the local peer.
		• This is the same key you just specified at the local peer.
		• If the local peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

Step 7	Repeat these steps at each peer that uses preshared keys in an IKE policy.	--

Configuring IKE Mode Configuration

Note IKE mode configuration has the following restrictions:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** *pool-name start-addr end-addr*
4. **crypto isakmp client configuration address-pool local** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** *pool-name start-addr end-addr* **Example:** `Router(config)# ip local pool pool1 172.16.23.0 172.16.23.255`	Defines an existing local address pool that defines a set of addresses.
Step 4	**crypto isakmp client configuration address-pool local** *pool-name* **Example:** `Router(config)# crypto isakmp client configuration address-pool local pool1`	References the local address pool in the IKE configuration.

Configuring an IKE Crypto Map for IPsec SA Negotiation

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *tag sequence* **ipsec-isakmp**
4. **set pfs** {**group1** | **group2** | **group5** | **group14** | **group15** | **group16**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *tag sequence* **ipsec-isakmp** **Example:** `Router(config)# crypto map example 1 ipsec-ipsec-isakmp`	Specifies the crypto map and enters crypto map configuration mode. • The *tag* argument specifies the crypto map. • The *sequence* argument specifies the sequence to insert into the crypto map entry. • The **ipsec-isakmp** keyword specifies IPsec with IKEv1 (ISAKMP).
Step 4	**set pfs** {**group1** | **group2** | **group5** | **group14** | **group15** | **group16**} **Example:** `Router(config-isakmp)# set pfs 14`	Specifies the DH group identifier for IPSec SA negotiation. • By default, DH group 1 is used. • **group1**—768-bit DH (No longer recommended) • **group2**—1024-bit DH (No longer recommended) • **group5**—1536-bit DH (No longer recommended) • **group14**—Specifies the 2048-bit DH group. • **group15**—Specifies the 3072-bit DH group. • **group16**—Specifies the 4096-bit DH group. The group chosen must be strong enough (have enough bits) to protect the IPsec keys during negotiation. A generally accepted guideline recommends the use of a 2048-bit group after 2013 (until 2030). Either group 14 can be selected to meet this guideline. Even if a longer-lived security method is needed, the use of Elliptic Curve Cryptography is recommended, but group 15 and group 16 can also be considered.

Configuration Examples for an IKE Configuration

Example: Creating IKE Policies

This section contains the following examples, which show how to configure an AES IKE policy and a 3DES IKE policy.

Note Cisco no longer recommends using 3DES; instead, you should use AES. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Example: Creating 3DES IKE Policies

This example creates two IKE policies, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
!
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
!
crypto isakmp key 1234567890 address 192.168.224.33
```

In the example, the encryption DES of policy default would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the **show crypto isakmp policy**command is issued with this configuration, the output is as follows:

```
Protection suite priority 15
encryption algorithm:3DES - Triple Data Encryption Standard (168 bit keys)
hash algorithm:Message Digest 5
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#2 (1024 bit)
lifetime:5000 seconds, no volume limit
Protection suite priority 20
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:preshared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:10000 seconds, no volume limit
Default protection suite
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#1 (768 bit)
lifetime:86400 seconds, no volume limit
```

Note that although the output shows “no volume limit” for the lifetimes, you can configure only a time lifetime (such as 86,400 seconds); volume-limit lifetimes are not configurable.

Example: Creating an AES IKE Policy

The following example is sample output from the **show running-config**command. In this example, the AES 256-bit key is enabled.

```
Current configuration : 1665 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname "Router1"
!
```

```
!
ip subnet-zero
!
!
no ip domain lookup
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport

.
.
.
```

Example: Configuring IKE Authentication

The following example shows how to manually specify the RSA public keys of two IPsec peer-- the peer at 10.5.5.1 uses general-purpose keys, and the other peer uses special-usage keys:

```
crypto key pubkey-chain rsa
 named-key otherpeer.example.com
 address 10.5.5.1
 key-string
 005C300D 06092A86 4886F70D 01010105
 00034B00 30480241 00C5E23B 55D6AB22
 04AEF1BA A54028A6 9ACC01C5 129D99E4
 64CAB820 847EDAD9 DF0B4E4C 73A05DD2
 BD62A8A9 FA603DD2 E2A8A6F8 98F76E28
 D58AD221 B583D7A4 71020301 0001
 quit
 exit
 addressed-key 10.1.1.2 encryption
 key-string
 00302017 4A7D385B 1234EF29 335FC973
 2DD50A37 C4F4B0FD 9DADE748 429618D5
 18242BA3 2EDFBDD3 4296142A DDF7D3D8
 08407685 2F2190A0 0B43F1BD 9A8A26DB
 07953829 791FCDE9 A98420F0 6A82045B
 90288A26 DBC64468 7789F76E EE21
 quit
 exit
 addressed-key 10.1.1.2 signature
 key-string
 0738BC7A 2BC3E9F0 679B00FE 53987BCC
 01030201 42DD06AF E228D24C 458AD228
 58BB5DDD F4836401 2A2D7163 219F882E
 64CE69D4 B583748A 241BED0F 6E7F2F16
 0DE0986E DF02031F 4B0B0912 F68200C4
 C625C389 0BFF3321 A2598935 C1B1
 quit
 exit
 exit
```

Where to Go Next

After you have successfully configured IKE negotiation, you can begin configuring IPsec. For information on completing these tasks, see the module "Configuring Security for VPNs With IPsec."

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	Configuring Security for VPNs with IPsec
IKE Version 2	Configuring Internet Key Exchange Version 2 and FlexVPN
Configuring RSA keys to obtain certificates from a CA	Deploying RSA Keys Within a PKI
Suite-B ESP transforms	Configuring Security for VPNs with IPsec
Suite-B Integrity algorithm type transform configuration.	Configuring Internet Key Exchange Version 2 and FlexVPN
Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	Configuring Internet Key Exchange Version 2 and FlexVPN
Suite-B support for certificate enrollment for a PKI	Configuring Certificate Enrollment for a PKI
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)
RFC 2409	The Internet Key Exchange (IKE)
RFC 2412	The OAKLEY Key Determination Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IKE for IPsec VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 264: Feature Information for Configuring IKE for IPsec VPNs

Feature Name	Releases	Feature Information
Ability to Disable Extended Authentication for Static IPsec Peers	12.2(4)T	This feature allows a user to disable Xauth while configuring the preshared key for router-to-router IPsec. Thus, the router will not prompt the peer for a username and password, which are transmitted when Xauth occurs for VPN-client-to-Cisco-IOS IPsec. The following command was modified by this feature: **crypto isakmp key.**
Advanced Encryption Standard (AES)	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPsec and IKE and has been developed to replace DES. The following commands were modified by this feature: **crypto ipsec transform-set, encryption (IKE policy), show crypto ipsec transform-set, crypto ipsec transform-set, show crypto isakmp policy.**
SEAL Encryption	12.3(7)T	This feature adds support for SEAL encryption in IPsec. The following command was modified by this feature: **crypto ipsec transform-set.**
Suite-B support in IOS SW crypto	15.1(2)T	Suite-B adds support in the Cisco IOS for the SHA-2 family (HMAC variant) hash algorithm used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. This feature also adds elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support. The following command was modified by this feature: **authentication, crypto key generate ec keysize, crypto map, group, hash, set pfs.**

CHAPTER 197

Call Admission Control for IKE

The Call Admission Control for IKE feature describes the application of Call Admission Control (CAC) to the Internet Key Exchange (IKE) protocol in Cisco IOS software. CAC limits the number of simultaneous IKE and IPsec security associations (SAs) that is, calls to CAC that a router can establish.

Prerequisites for Call Admission Control for IKE

- Configure IKE on the device.

Information About Call Admission Control for IKE

IKE Session

There are two ways to limit the number of Internet Key Exchange (IKE) security associations (SAs) that a device can establish to or from another device:

- Configure the absolute IKE SA limit by entering the **crypto call admission limit** command. The device drops new IKE SA requests when the configured limit is reached.
- Configure the system resource limit by entering the **call admission limit** command. The device drops new IKE SA requests when the level of system resources that are configured in the unit of charge is being used.

Call Admission Control (CAC) is applied only to new SAs (that is, when an SA does not already exist between peers). Every effort is made to preserve existing SAs. New SA requests are denied due to a lack of system resources or because the configured IKE SA limit is reached.

Security Association Limit

An SA is a description of how two or more entities will utilize security services to communicate securely on behalf of a particular data flow. IKE requires and uses SAs to identify the parameters of its connections. IKE can negotiate and establish its own SA. An IKE SA is used by IKE only, and it is bidirectional. An IKE SA cannot limit IPsec.

IKE drops SA requests based on a user-configured SA limit. To configure an IKE SA limit, enter the **crypto call admission limit** command. When there is a new SA request from a peer router, IKE determines whether the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.

The **ipsec sa** *number* and **ike sa** *number* keyword and argument pairs in the **crypto call admission limit**command set the limit for the number of established IPsec SAs and IKE SAs.

Limit on Number of In-Negotiation IKE Connections

You can limit the number of in-negotiation IKE connections that can be configured on a device based on your Cisco release. This type of IKE connection represents either an aggressive mode IKE SA or a main mode IKE SA prior to its authentication and actual establishment. The default value for maximum in-negotiation CAC for IKEv2 is 40.

You can use the **crypto call admission limit ike in-negotiation-sa** *number* command to specify the maximum number of Internet Key Exchange (IKE) and IPsec security associations (SAs) that the device can establish before IKE begins rejecting the new SA requests.

The **all in-negotiation-sa** *number* and **ike in-negotiation-sa** *number* keyword and argument pairs in the **crypto call admission limit** command limit all SAs in negotiation and IKE SAs in negotiation.

System Resource Usage

CAC polls a global resource monitor so that IKE knows when the router is running short of CPU cycles or memory buffers. You can configure a limit, in the range 1 to 100000, that represents the level of system resource usage in system resource usage units. When that level of resources is being used, IKE drops (will not accept new) SA requests. To configure the system resource usage limit, enter the **call admission limit** command.

For each incoming new SA request, the current load on the router is converted into a numerical value, representing the system resource usage level, and is compared to the resource limit set by the **call admission limit** command. If the current load is more than the configured resource limit, IKE drops the new SA request. Load on the router includes active SAs, CPU usage, and SA requests being considered.

The **call admission load** command configures a multiplier value from 0 to 1000 that represents a scaling factor for current system resource usage and a load metric poll rate of 1 to 32 seconds. The numerical value for the system resource usage level is calculated by the formula (scaling factor * current system resource usage) / 100. It is recommended that the **call admission load** command not be used unless advised by a Cisco Technical Assistance Center (TAC) engineer.

How to Configure Call Admission Control for IKE

Configuring the IKE Security Association Limit

Perform this task to configure the absolute IKE SA limit. The router drops new IKE SA requests when the limit has been reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto call admission limit** {**all in-negotiation-sa** *number* | **ipsec sa** *number* | **ike** {**in-negotiation-sa** *number* | **sa** *number*}}
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto call admission limit** {**all in-negotiation-sa** *number*	**ipsec sa** *number*	**ike** {**in-negotiation-sa** *number*	**sa** *number*}} **Example:** `Router(config)# crypto call admission limit ike sa 25`	Specifies the maximum number of IKE SAs or total SAs in negotiation or the maximum IKE SAs or IPsec SAs that can be established before IKE begins rejecting new SA requests. The recommended CAC value for IKEv1 is 40.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the IKEv2 Security Association Limit

Perform this task to configure the absolute IKEv2 SA limit. The router drops new IKE SA requests when the limit has been reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 limit** {**max-in-negotiation-sa limit** *number* | **max-sa limit** *number*}
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 limit** {**max-in-negotiation-sa limit** *number* | **max-sa limit** *number*} **Example:** `Router(config)# crypto ikev2 limit max-in-negotation-sa 5000`	Enables call admission control as follows: • max-in-negotation-sa limit: Limits the total number of in-negotiation IKEv2 SAs on the node. • max-sa limit: Limits the total number of IKEv2 SAs on the node.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the System Resource Limit

Perform this task to configure the system resource limit. The router drops new IKE SA requests when the level of system resources that are configured in the unit of charge is being used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call admission limit** *charge*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call admission limit** *charge* **Example:** `Router(config)# call admission limit 1000`	Sets the level of the system resources that, when used, causes IKE to stop accepting new SA requests. • *charge* --Valid values are 1 to 100000.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the Call Admission Control for IKE Configuration

To verify the CAC for IKE configuration, perform the following steps.

SUMMARY STEPS

1. **show call admission statistics**
2. **show crypto call admission statistics**

DETAILED STEPS

Step 1 **show call admission statistics**

Use this command to monitor the global CAC configuration parameters and the behavior of CAC.

Example:

```
Router# show call admission statistics
Total Call admission charges: 82, limit 1000
Total calls rejected 1430, accepted 0
Load metric: charge 82, unscaled 82%
```

Step 2 **show crypto call admission statistics**

Use this command to monitor crypto CAC statistics.

Example:

```
Router# show crypto call admission statistics
----------------------------------------------------------------------
               Crypto Call Admission Control Statistics
----------------------------------------------------------------------
System Resource Limit:      111 Max IKE SAs:     0 Max in nego:  1000
Total IKE SA Count:           0 active:          0 negotiating:     0
Incoming IKE Requests:        0 accepted:        0 rejected:        0
Outgoing IKE Requests:        0 accepted:        0 rejected:        0
Rejected IKE Requests:        0 rsrc low:        0 Active SA limit: 0
                                                   In-neg SA limit: 0
IKE packets dropped at dispatch:        0
Max IPSEC SAs:   111
Total IPSEC SA Count:           0 active:          0 negotiating:     0
Incoming IPSEC Requests:        0 accepted:        0 rejected:        0
Outgoing IPSEC Requests:        0 accepted:        0 rejected:        0
Phase1.5 SAs under negotiation:         0
```

Configuration Examples for Call Admission Control for IKE

Example Configuring the IKE Security Association Limit

The following example shows how to specify a maximum limit of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

Example Configuring the System Resource Limit

The following example shows how to specify that IKE should drop SA requests when the level of system resources that are configured in the unit of charge reaches 9000:

```
Router(config)# call admission limit 9000
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs
IKE commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2409	*The Internet Key Exchange*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Call Admission Control for IKE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 265: Feature Information for Call Admission Control for IKE

Feature Name	Releases	Feature Information
Call Admission Control for IKE	12.3(8)T 12.2(18)SXD1 12.4(6)T 12.2(33)SRA 12.2(33)SXH	The Call Admission Control for IKE feature describes the application of Call Admission Control (CAC) to the Internet Key Exchange (IKE) protocol in Cisco IOS software. In Cisco IOS Release 12.3(8)T, this feature was introduced. This feature was integrated into Cisco IOS Release 12.2(18)SXD1 and implemented on the Cisco 6500 and Cisco 7600 routers. In Cisco IOS Release 12.4(6)T, the ability to configure a limit on the number of in-negotiation IKE connections was added. The following sections provide information about this feature: The following commands were introduced or modified: **call admission limit**, **clear crypto call admission statistics**, **crypto call admission limit**, **show call admission statistics**, **show crypto call admission statistics**.
IKEv1 Hardening	15.1(3)T	The IKEv1 hardening feature describes the enhancements made to the Call Admission Control (CAC) for IKE feature. In Cisco IOS Release 15.1(3)T, this feature was introduced. The following sections provide information about this feature: The following commands were introduced or modified: **crypto call admission limit**, **show crypto call admission statistics**.

CHAPTER 198

Certificate to ISAKMP Profile Mapping

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

Prerequisites for Certificate to ISAKMP Profile Mapping

- You should be familiar with configuring certificate maps.
- You should be familiar with configuring ISAKMP profiles.

Restrictions for Certificate to ISAKMP Profile Mapping

This feature is not applicable if you use Rivest, Shamir, and Adelman (RSA)-signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured for RSA-signature or RSA-encryption authentication using certificates.

IPsec with two trustpoints enrolled in the same Certificate Authority (CA) server is not supported. When there are two or more ISAKMP profiles, each having a different trustpoint enrolled in the same CA server, the responder selects the last global trustpoint. (Trustpoints are selected in the reverse order in which they are defined globally). For the IPsec tunnel establishment to be successful for peers, the trustpoint selected by the initiator should match the trustpoint selected by the responder. All other IPsec tunnels will fail to establish connection if the trustpoints do not match.

Information About Certificate to ISAKMP Profile Mapping

Certificate to ISAKMP Profile Mapping Overview

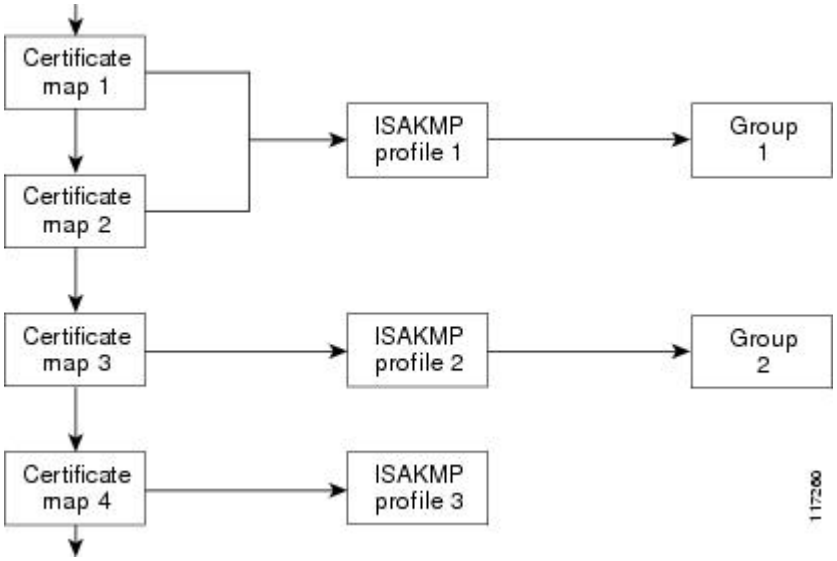
Prior to Cisco IOS Release 12.3(8)T, the only way to map a peer to an ISAKMP profile was as follows. The ISAKMP identity field in the ISAKMP exchange was used for mapping a peer to an ISAKMP profile. When certificates were used for authentication, the ISAKMP identity payload contained the subject name from the certificate. If a CA did not provide the required group value in the first Organizational Unit (OU) field of a certificate, an ISAKMP profile could not be assigned to a peer.

Effective with Cisco IOS Release 12.3(8)T, a peer can still be mapped as explained above. However, the Certificate to ISAKMP Profile Mapping feature enables you to assign an ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate. You are no longer limited to assigning an ISAKMP profile on the basis of the subject name of the certificate. In addition, this feature allows you to assign a group to a peer to which an ISAKMP profile has been assigned.

How Certificate to ISAKMP Profile Mapping Works

The figure below illustrates how certificate maps may be attached to ISAKMP profiles and assigned group names.

Figure 100: Certificate Maps Mapped for Profile Group Assignment



A certificate map can be attached to only one ISAKMP profile although an ISAKMP profile can have several certificate maps attached to it.

Certificate maps provide the ability for a certificate to be matched with a given set of criteria. ISAKMP profiles can bind themselves to certificate maps, and if the presented certificate matches the certificate map present in an ISAKMP profile, the peer will be assigned the ISAKMP profile. If the ISAKMP profile contains a client configuration group name, the same group name will be assigned to the peer. This ISAKMP profile information will override the information in the ID_KEY_ID identity or in the first OU field of the certificate.

Assigning an ISAKMP Profile and Group Name to a Peer

To assign an ISAKMP profile to a peer on the basis of arbitrary fields in the certificate, use the **match certificate** command after the ISAKMP profile has been defined.

To associate a group name with an ISAKMP profile that will be assigned to a peer, use the **client configuration group** command, also after the ISAKMP profile has been defined.

How to Configure Certificate to ISAKMP Profile Mapping

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following steps. This configuration will enable you to assign the ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **match certificate** *certificate-map*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into crypto ISAKMP profile configuration mode.
Step 4	**match certificate** *certificate-map* **Example:** `Router (conf-isa-prof)# match certificate map1`	Accepts the name of a certificate map.

Verifying That the Certificate Has Been Mapped

The following **show** command may be used to verify that the subject name of the certificate map has been properly configured.

SUMMARY STEPS

1. **enable**
2. **show crypto ca certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ca certificates** **Example:** `Router# show crypto ca certificates`	Displays information about your certificate.

Assigning the Group Name to the Peer

To associate a group name with a peer when the peer is mapped to an ISAKMP profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **client configuration group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into isakmp profile configuration mode.
Step 4	**client configuration group** *group-name* **Example:** `Router (conf-isa-prof)# client configuration group group1`	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile.

Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping

To monitor and maintain your certificate to ISAKMP profile mapping, you may use the following **debug** command.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays output showing that the certificate has gone through certificate map matching and that the certificate matches the ISAKMP profile. The command may also be used to verify that the peer has been assigned a group.

Configuration Examples for Certificate to ISAKMP Profile Mapping

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Example

The following configuration example shows that whenever a certificate contains “ou = green,” the ISAKMP profile “cert_pro” will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   initiate mode aggressive
   match certificate cert_map
```

Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile Example

The following example shows that the group “some_group” is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
   match identity host domain cisco.com
   client configuration group some_group
```

Mapping a Certificate to an ISAKMP Profile Verification Example

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, **show command** output verifying that the subject name of the certificate map has been configured, and **debug** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
 subject-name co ou = green
! The above line shows that the subject name must have “ou = green.”
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
```

```
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   initiate mode aggressive
```

Initiator Configuration

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name "ou = green" is set.
 revocation-check none
```

show crypto ca certificates Command Output for the Initiator

```
Router# show crypto ca certificates
Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router1.cisco.com
    c=IN
    ou=green
! The above line is a double check that "ou = green" has been set as the subject name.
    hostname=Router1.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

debug crypto isakmp Command Output for the Responder

```
Router# debug crypto isakmp
6d23h: ISAKMP (0:268435460): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
 MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:4:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:4:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:4:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435460): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:4:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:4:HW:2): processing CERT payload. message ID = 0
```

```
6d23h: ISAKMP:(0:4:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:4:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:4:HW:2): OU = green
6d23h: ISAKMP:(0:4:HW:2): certificate map matches certpro profile
! The above line shows that the certificate has gone through certificate map matching and
that it matches the "certpro" profile.
6d23h: ISAKMP:(0:4:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:4:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:4:HW:2): CERT validity confirmed.
```

Group Name Assigned to a Peer Verification Example

The following configuration and debug output show that a group has been assigned to a peer.

Initiator Configuration

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group "new_group" to any peer that matches
 the ISAKMP profile "certpro."
   initiate mode aggressive
!
```

debug crypto isakmp profile Command Output for the Responder

The following debug output example shows that the peer has been matched to the ISAKMP profile named "certpro" and that it has been assigned a group named "new_group."

```
Router# debug crypto isakmp profile
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
 MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
```

```
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

Additional References

Related Documents

Related Topic	Document Title
Configuring ISAKMP profiles	VRF-Aware IPsec
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Certificate to ISAKMP Profile Mapping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 266: Feature Information for Certificate to ISAKMP Profile Mapping

Feature Name	Releases	Feature Information
Certificate to ISAKMP Profile Mapping	12.3(8)T 12.2(33)SRA 12.2(33)SXH	The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile. This feature was introduced in the Cisco IOS Release 12.3(8)T This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH.

CHAPTER 199

Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Restrictions for Encrypted Preshared Key

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. Therefore, errors are expected if you boot from an old ROMMON.
- For Cisco 836 routers, please note that support for Advanced Encryption Standard (AES) is available only on IP plus images.

Information About Encrypted Preshared Key

Using the Encrypted Preshared Key Feature to Securely Store Passwords

Using the Encrypted Preshared Key feature, you can securely store plain text passwords in type 6 format in NVRAM using a command-line interface (CLI). Type 6 passwords are encrypted. Although the encrypted passwords can be seen or retrieved, it is difficult to decrypt them to find out the actual password. Use the **key config-key** command with the **password encryption aes** command to configure and enable the password (symmetric cipher AES is used to encrypt the keys). The password (key) configured using the **config-key password-encryption** command is the master encryption key that is used to encrypt all other keys in the router.

If you configure the **password encryption aes**command without configuring the **key config-key** command, the following message is printed at startup or during any nonvolatile generation (NVGEN) process, such as when the **show running-config** or **copy running-config startup-config** commands have been configured:

```
"Can not encrypt password. Please configure a configuration-key with 'key config-key'"
```

Changing a Password

If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption**command), the list registry passes the old key and the new key to the application modules that are using type 6 encryption.

Deleting a Password

If the master key that was configured using the **key config-key password-encryption**command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the software. However, passwords can be reencrypted as explained in the previous paragraph.

Caution If the password configured using the **key config-key password-encryption**command is lost, it cannot be recovered. The password should be stored in a safe location.

Unconfiguring Password Encryption

If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption**command exists, the type 6 passwords will be decrypted as and when required by the application.

Storing Passwords

Because no one can "read" the password (configured using the **key config-key password-encryption**command), there is no way that the password can be retrieved from the router. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a router. Before or after the configurations are loaded onto a router, the password must be manually added (using the **key config-key password-encryption**command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

Configuring New or Unknown Passwords

If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but an alert message is printed. The alert message is as follows:

```
"ciphertext>[for username bar>] is incompatible with the configured master key."
```

If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.

If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption**command. Deleting the master key using the **no key config-key**

password-encryptioncommand causes the existing encrypted passwords to remain encrypted in the router configuration. The passwords will not be decrypted.

Enabling the Encrypted Preshared Key

The **password encryption aes** command is used to enable the encrypted password.

How to Configure an Encrypted Preshared Key

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key password-encryption** [*text*]
4. **password encryption aes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key config-key password-encryption** [*text*] **Example:** `Router (config)# key config-key password-encryption`	Stores a type 6 encryption key in private NVRAM. • If you want to key in interactively (using the enter key) and an encrypted key already exists, you will be prompted for the following: Old key, New key, and Confirm key. • If you want to key in interactively but an encryption key is not present, you will be prompted for the following: New key and Confirm key. • If you want to remove the password that is already encrypted, you will see the following prompt: "WARNING: All type 6 encrypted keys will become

	Command or Action	Purpose
		unusable. Continue with master key deletion? [yes/no]:”.
Step 4	**password encryption aes** **Example:** `Router (config)# password-encryption aes`	Enables the encrypted preshared key.

Troubleshooting Tips

If you see the warning message “ciphertext >[for username bar>] is incompatible with the configured master key,” you have entered or cut and pasted cipher text that does not match the master key or there is no master key. (The cipher text will be accepted or saved.) The warning message will allow you to locate the broken configuration line or lines.

Monitoring Encrypted Preshared Keys

To get logging output for encrypted preshared keys, perform the following steps.

1. **enable**
2. **password logging**

SUMMARY STEPS

1. **enable**
2. **password logging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**password logging** **Example:** `Router# password logging`	Provides a log of debugging output for a type 6 password operation.

Examples

The following **password logging** debug output shows that a new master key has been configured and that the keys have been encrypted with the new master key:

```
Router (config)# key config-key password-encrypt
New key:
Confirm key:
Router (config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas
Router (config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

What To Do Next

You can perform any of the following procedures. Each procedure is independent of the others.

Configuring an ISAKMP Preshared Key

To configure an ISAKMP preshared key, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp key** *keystring* **address** *peer-address*
4. **crypto isakmp key** *keystring* **hostname** *hostname*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp key** *keystring* **address** *peer-address* **Example:** `Router (config)# crypto isakmp key cisco address 10.2.3.4`	Configures a preshared authentication key. • The *peer-address* argument specifies the IP address of the remote peer.

	Command or Action	Purpose
Step 4	**crypto isakmp key** *keystring* **hostname** *hostname* **Example:** `Router (config)# crypto isakmp key mykey hostname mydomain.com`	Configures a preshared authentication key. • The *hostname* argument specifies the fully qualified domain name (FQDN) of the peer.

Example

The following sample output shows that an encrypted preshared key has been configured:

```
crypto isakmp key 6 _Hg[^^ECgLGGPF^RXTQfDDWQ][YAAB address 10.2.3.4
crypto isakmp key 6 `eR\eTRaKCUZPYYQfDgXRWi_AAB hostname mydomain.com
```

Configuring an ISAKMP Preshared Key in ISAKMP Keyrings

To configure an ISAKMP preshared key in ISAKMP keyrings, which are used in IPSec Virtual Route Forwarding (VRF) configurations, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name*
4. **pre-shared-key address** *address* **key** *key*
5. **pre-shared-key hostname** *hostname* **key** *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* **Example:** `Router (config)# crypto keyring mykeyring`	Defines a crypto keyring to be used during Internet Key Exchange (IKE) authentication and enters keyring configuration mode.
Step 4	**pre-shared-key address** *address* **key** *key*	Defines a preshared key to be used for IKE authentication.

	Command or Action	Purpose
	Example: `Router (config-keyring)# pre-shared-key address 10.2.3.5 key cisco`	• The *address* argument specifies the IP address of the remote peer.
Step 5	**pre-shared-key hostname** *hostname* **key** *key* **Example:** `Router (config-keyring)# pre-shared-key hostname mydomain.com key cisco`	Defines a preshared key to be used for IKE authentication. • The *hostname* argument specifies the FQDN of the peer.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP keyrings has been configured.

```
crypto keyring mykeyring
  pre-shared-key address 10.2.3.5 key 6 `WHCJYR_Z]GRPF^RXTQfDcfZ]GPAAB
  pre-shared-key hostname mydomain.com key 6 aE_REHDcOfYCPF^RXTQfDJYVVNSAAB
```

Configuring ISAKMP Aggressive Mode

To configure ISAKMP aggressive mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer ip-address** *ip-address*
4. **set aggressive-mode client-endpoint** *client-endpoint*
5. **set aggressive-mode password** *password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto isakmp peer ip-address** *ip-address* **Example:** `Router (config)# crypto isakmp peer ip-address 10.2.3.4`	To enable an IP Security (IPSec) peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and to enter ISAKMP peer configuration mode.
Step 4	**set aggressive-mode client-endpoint** *client-endpoint* **Example:** `Router (config-isakmp-peer)# set aggressive-mode client-endpoint fqdn cisco.com`	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.
Step 5	**set aggressive-mode password** *password* **Example:** `Router (config-isakmp-peer)# set aggressive-mode password cisco`	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP aggressive mode has been configured.

```
crypto isakmp peer address 10.2.3.4
 set aggressive-mode password 6 ^aKPIQ_KJE_PPF^RXTQfDTIaLNeAAB
 set aggressive-mode client-endpoint fqdn cisco.com
```

Configuring a Unity Server Group Policy

To configure a unity server group policy, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **pool** *name*
5. **domain name**
6. **key** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router# enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group mygroup`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode.
Step 4	**pool** *name* **Example:** `Router (config-isakmp-group)# pool mypool`	Defines a local pool address.
Step 5	**domain name** **Example:** `Router (config-isakmp-group)# domain cisco.com`	Specifies the Domain Name Service (DNS) domain to which a group belongs.
Step 6	**key** *name* **Example:** `Router (config-isakmp-group)# key cisco`	Specifies the IKE preshared key for group policy attribute definition.

Example

The following **show-running-config** sample output shows that an encrypted key has been configured for a unity server group policy:

```
crypto isakmp client configuration group mygroup
 key 6 cZZgDZPOE\dDPF^RXTQfDTIaLNeAAB
 domain cisco.com
 pool mypool
```

Configuring an Easy VPN Client

To configure an Easy VPN client, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto ipsec client ezvpn** *name*
4. **peer** *ipaddress*
5. **mode client**
6. **group** *group-name* **key** *group-key*
7. **connect manual**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn myclient`	Creates a Cisco Easy VPN remote configuration and enters Cisco Easy VPN remote configuration mode.
Step 4	**peer** *ipaddress* **Example:** `Router (config-isakmp-peer)# peer 10.2.3.4`	Sets the peer IP address for the VPN connection.
Step 5	**mode client** **Example:** `Router (config-isakmp-ezpvy)# mode client`	Automatically configures the router for Cisco Easy VPN Client mode operation, which uses Network Address Translation (NAT) or Peer Address Translation (PAT) address translations.
Step 6	**group** *group-name* **key** *group-key* **Example:** `Router (config-isakmp-ezvpn)# group mygroup key cisco`	Specifies the group name and key value for the VPN connection.
Step 7	**connect manual** **Example:** `Router (config-isakmp-ezvpn)# connect manual`	Specifies the manual setting for directing the Cisco Easy VPN remote client to wait for a command or application program interface (API) call before attempting to establish the Cisco Easy VPN remote connection.

Example

The following **show-running-config** sample output shows that an Easy VPN client has been configured. The key has been encrypted.

```
crypto ipsec client ezvpn myclient
 connect manual
 group mygroup key 6 gdMI`S^^[GIcPF^RXTQfDFKEO\RAAB
 mode client
 peer 10.2.3.4
```

Configuration Examples for Encrypted Preshared Key

Encrypted Preshared Key Example

The following is an example of a configuration for which a type 6 preshared key has been encrypted. It includes the prompts and messages that a user might see.

```
Router (config)# crypto isakmp key cisco address 10.0.0.2
Router (config)# exit
Router# show running-config | include crypto isakmp key
 crypto isakmp key cisco address 10.0.0.2
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router (config)# password encryption aes

Router (config)# key config-key password-encrypt

New key:
Confirm key:
Router (config)#
01:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router (config)# exit
Router # show running-config | include crypto isakmp key
crypto isakmp key 6 CXWdhVTZYB_Vcd^`cIHDOahiFTa address 10.0.0.2
```

No Previous Key Present Example

In the following configuration example, no previous key is present:

```
Router (config)#
```

Key Already Exists Example

In the following configuration example, a key already exists:

```
Router (config)#
```

```
Old key:
Router (config)#
```

Key Already Exists But the User Wants to Key In Interactively Example

In the following configuration example, the user wants to key in interactively, but a key already exists. The Old key, New key, and Confirm key prompts will show on your screen if you enter the **key config-key** command and press the enter key to get into interactive mode.

```
Router (config)#
Old key:
New key:
Confirm key:
```

No Key Present But the User Wants to Key In Interactively Example

In the following example, the user wants to key in interactively, but no key is present. The New key and Confirm key prompts will show on your screen if you are in interactive mode.

```
Router (config)#

New key:
Confirm key:
```

Removal of the Password Encryption Example

In the following configuration example, the user wants to remove the encrypted password. The "WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:" prompt will show on your screen if you are in interactive mode.

```
Router (config)#
WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion
 ? [yes/no]: y
```

Where to Go Next

Configure any other preshared keys.

Additional References

Related Documents

Related Topic	Document Title
Configuring passwords	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 200

Distinguished Name Based Crypto Maps

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

This feature module describes the Distinguished Name Based Crypto Map feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

Feature Overview

The Distinguished Name Based Crypto Maps feature allows you to configure the router to restrict access to selected encrypted interfaces for those peers with specific certificates, especially certificates with particular Distinguished Names (DNs).

Previously, if the router accepted a certificate or a shared secret from the encrypting peer, Cisco IOS did not have a method of preventing the peer from communicating with any encrypted interface other than the restrictions on the IP address of the encrypting peer. This feature allows you to configure which crypto maps are usable to a peer based on the DN that a peer used to authenticate itself, thereby, enabling you to control which encrypted interfaces a peer with a specified DN can access.

Benefits

The Distinguished Name Based Crypto Maps feature allows you to set restrictions in the router configuration that prevent peers with specific certificates--especially certificates with particular DNs-- from having access to selected encrypted interfaces.

Restrictions

System Requirements

To configure this feature, your router must support IP Security.

Performance Impact

If you restrict access to a large number of DNs, it is recommended that you specify a few number of crypto maps referring to large identity sections instead of specifying a large number of crypto maps referring to small identity sections.

Related Documents

The following documents provide information related to the Distinguished Name Based Crypto Maps feature:

- Cisco IOS Security Command Reference
- Cisco IOS Security Configuration Guide: Secure Connectivity, Release 12.4T
- Next Generation Encryption (NGE) white paper.

Supported Platforms

This feature is supported on the following platforms:

- Cisco 1700 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco uBR905 Cable Access Router
- Cisco uBR925 Cable Access Router

Determining Platform Support Through Feature Navigator

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Supported Standards MIBs and RFCs

Standards

None

MIBs

None

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

RFCs

None

Prerequisites

Before configuring a DN based crypto map, you must perform the following tasks:

- Create an Internet Key Exchange (IKE) policy at each peer.

For more information on creating IKE policies, refer to the " Configuring Internet Key Exchange for IPsec VPNs " chapter in the *Cisco IOS Security Configuration Guide: Secure Connectivity* ..

- Create crypto map entries for IPSec.

For more information on creating crypto map entries, refer to the " Configuring Security for VPNs with IPsec " chapter in the *Cisco IOS Security Configuration Guide: Secure Connectivity*

Configuration Tasks

See the following sections for configuration tasks for the Distinguished Name Based Crypto Maps feature. Each task in the list is identified as either required or optional.

- Configuring DN Based Crypto Maps (authenticated by DN), on page 2588 (required)
- Configuring DN Based Crypto Maps (authenticated by hostname), on page 2588 (required)
- Applying Identity to DN Based Crypto Maps, on page 2589 (required)
- Verifying DN Based Crypto Maps, on page 2589 (optional)

Configuring DN Based Crypto Maps (authenticated by DN)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a DN, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **crypto identity** *name*
2. Router(crypto-identity)# **dn** *name=string* [,*name=string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **crypto identity** *name*	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	Router(crypto-identity)# **dn** *name=string* [,*name=string*]	Associates the identity of the router with the DN in the certificate of the router. **Note** The identity of the peer must match the identity in the exchanged certificate.

Configuring DN Based Crypto Maps (authenticated by hostname)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a hostname, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **crypto identity** *name*
2. Router(crypto-identity)# **fqdn** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **crypto identity** *name*	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	Router(crypto-identity)# **fqdn** *name*	Associates the identity of the router with the hostname that the peer used to authenticate itself. **Note** The identity of the peer must match the identity in the exchanged certificate.

Applying Identity to DN Based Crypto Maps

To apply the identity (within the crypto map context), use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**
2. Router(config-crypto-map)# **identity** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
Step 2	Router(config-crypto-map)# **identity** *name*	Applies the identity to the crypto map. When this command is applied, only the hosts that match a configuration listed within the **identity** *name* can use the specified crypto map. **Note** If the **identity** command does not appear within the crypto map, the encrypted connection does not have any restrictions other than the IP address of the encrypting peer.

Verifying DN Based Crypto Maps

To verify that this functionality is properly configured, use the following command in EXEC mode:

Command	Purpose
`Router# show crypto identity`	Displays the configured identities.

Troubleshooting Tips

If an encrypting peer attempts to establish a connection that is blocked by the DN based crypto map configuration, the following error message will be logged:

```
<time>: %CRYPTO-4-IKE_QUICKMODE_BAD_CERT: encrypted connection attempted with a peer without
 the configured certificate attributes.
```

Configuration Examples

DN Based Crypto Map Configuration Example

The following example shows how to configure DN based crypto maps that have been authenticated by DN and hostname. Comments are included inline to explain various commands.

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
 encryption aes
 hash sha
 authentication rsa-sig
 group 14
 lifetime 5000
crypto isakmp policy 20
 encryption aes
 hash sha
 authentication pre-share
 group 14
 lifetime 10000
crypto isakmp key 1234567890 address 171.69.224.33
!
! The following is an IPSec crypto map (part of IPSec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
 dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
! and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
 fqdn little.com
!
```

CHAPTER 201

IPsec and Quality of Service

The IPsec and Quality of Service feature allows Cisco IOS quality of service (QoS) policies to be applied to IP Security (IPsec) packet flows on the basis of a QoS group that can be added to the current Internet Security Association and Key Management Protocol (ISAKMP) profile.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp . You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Prerequisites for IPsec and Quality of Service

- You should be familiar with IPsec and the concept of ISAKMP profiles.
- You should be familiar with Cisco IOS QoS.

Restrictions for IPsec and Quality of Service

- This feature can be applied only via the ISAKMP profile. The limit of 128 QoS groups that exists for QoS applications applies to this feature as well.
- You can apply an IPsec QoS group only to outbound service policies.
- QoS is not supported for software encryption.

Information About IPsec and Quality of Service

IPsec and Quality of Service Overview

The IPsec and Quality of Service feature allows you to apply QoS policies, such as traffic policing and shaping, to IPsec-protected packets by adding a QoS group to ISAKMP profiles. After the QoS group has been added, this group value will be mapped to the same QoS group as defined in QoS class maps. Any current QoS method that makes use of this QoS group tag can be applied to IPsec packet flows. Common groupings of packet flows can have specific policy classes applied by having the IPsec QoS group made available to the QoS mechanism. Marking IPsec flows allows QoS mechanisms to be applied to classes of traffic that could provide support for such things as restricting the amount of bandwidth that is available to specific groups or devices or marking the type of service (ToS) bits on certain flows.

The application of the QoS group is applied at the ISAKMP profile level because it is the profile that can uniquely identify devices through its concept of match identity criteria. These criteria are on the basis of the Internet Key Exchange (IKE) identity that is presented by incoming IKE connections and includes such things as IP address, fully qualified domain name (FQDN), and group (that is, the virtual private network [VPN] remote client grouping). The granularity of the match identity criteria will impose the granularity of the specified QoS policy, for example, to mark all traffic belonging to the VPN client group named “Engineering” as “TOS 5”. Another example of having the granularity of a specified QoS policy imposed would be to allocate 30 percent of the bandwidth on an outbound WAN link to a specific group of remote VPN devices.

How to Configure IPsec and Quality of Service

Configuring IPsec and Quality of Service

To apply QoS policies to an ISAKMP profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp-profile** *profile-number*
4. **qos-group** *group-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp-profile** *profile-number* **Example:** `Router (config)# crypto isakmp-profile vpnprofile`	Defines an ISAKMP profile, audits IPsec user sessions, and enters ISAKMP profile configuration mode.
Step 4	**qos-group** *group-number* **Example:** `Router(config-isa-prof)# qos-group 1`	Applies a QoS group value to an ISAKMP profile.

Verifying IPsec and Quality of Service Sessions

To verify your IPsec and QoS sessions, perform the following steps. The **show** commands can be used in any order or independent of each other.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp profile**
3. **show crypto ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Shows that the QoS group is applied to the profile.

	Command or Action	Purpose
Step 3	**show crypto ipsec sa** **Example:** `Router# show crypto ipsec sa`	Shows that the QoS group is applied to a particular pair of IPsec security associations (SAs).

Troubleshooting Tips

If you have a problem with your IPsec and QoS sessions, ensure that you have done the following:

- Validated the application of QoS by the QoS service using the QoS-specific commands in the *Cisco IOS Quality of Service Solutions Command Reference*.
- Configured a QoS policy on the router that matches the same QoS group as that specified for the class map match criterion.
- Applied the service policy to the same interface to which a crypto map is applied.

Configuration Examples for IPsec and Quality of Service

QoS Policy Applied to Two Groups of Remote Users Example

In the following example, a specific QoS policy is applied to two groups of remote users. Two ISAKMP profiles are configured so that upon initial connection via IKE, remote users are mapped to a specific profile. From that profile, all IPsec SAs that have been created for that remote will be marked with the specific QoS group. As traffic leaves the outbound interface, the QoS service will map the IPsec set QoS group with the QoS group that is specified in the class maps that comprise the service policy that is applied on that outbound interface.

```
version 12.3
!
aaa authentication login group group radius
aaa authorization network autho local
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
!
!
ip cef
no ip domain lookup
!
class-map match-all yellow
 match qos-group 3
class-map match-all blue
match qos-group 2
!
!
policy-map clients
 class blue
  set precedence 5
 class yellow
```

```
  set precedence 7
!
!
crypto isakmp policy 1
 encr aes
 hash sha
 authentication pre-share
 group 14
 lifetime 300
!
crypto isakmp keepalive 10 periodic
crypto isakmp xauth timeout 20
!
crypto isakmp client configuration group blue
 key cisco
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.6
 pool blue
 save-password
 include-local-lan
 backup-gateway corky1.cisco.com
!
crypto isakmp client configuration group yellow
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.5
 pool yellow
!
crypto isakmp profile blue
   match identity group cisco
   client authentication list autho
   isakmp authorization list autho
   client configuration address respond
   qos-group 2
crypto isakmp profile yellow
   match identity group yellow
   match identity address 10.0.0.11 255.255.255.255
   client authentication list autho
   isakmp authorization list autho
   client configuration address respond
   qos-group 3
!
!
crypto ipsec transform-set combo ah-sha-hmac esp-aes esp-sha-hmac
crypto ipsec transform-set client esp-aes esp-sha-hmac comp-lzs
!
crypto dynamic-map mode 1
 set security-association lifetime seconds 180
 set transform-set client
 set isakmp-profile blue
 reverse-route
crypto dynamic-map mode 2
 set transform-set combo
 set isakmp-profile yellow
 reverse-route
!
crypto map mode 1 ipsec-isakmp dynamic mode
!
interface FastEthernet0/0
 ip address 10.0.0.110 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip mroute-cache
 duplex half
 no cdp enable
```

```
 crypto map mode
 service-policy out clients
!
ip local pool yellow 192.168.2.1 192.168.2.10
ip local pool blue 192.168.6.1 192.168.6.6
no ip classless
!
radius-server host 10.0.0.13 auth-port 1645 acct-port 1646
radius-server key XXXXXX
radius-server vsa send accounting
radius-server vsa send authentication
```

show crypto isakmp profile Command Example

The following output shows that QoS group “2” has been applied to the ISAKMP profile “blue” and that QoS group “3” has been applied to the ISAKMP profile “yellow”:

```
Router# show crypto isakmp profile
ISAKMP PROFILE blue
   Identities matched are:
    group blue
   QoS Group 2 is applied
ISAKMP PROFILE yellow
   Identities matched are:
    ip-address 10.0.0.13 255.255.255.255
    group yellow
   QoS Group 3 is applied
```

show crypto ipsec sa Command Example

The following output shows that the QoS group has been applied to a particular pair of IPsec SAs:

```
Router# show crypto ipsec sa
interface: FastEthernet0/0
    Crypto map tag: mode, local addr. 10.0.0.110
   protected vrf:
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (10.12.12.0/255.255.255.0/0/0)
   current_peer: 10.0.0.11:500
     PERMIT, flags={}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     qos group is set to 2
```

Additional References

The following sections provide references related to the IPsec and Quality of Service feature.

Related Documents

Related Topic	Document Title
IPsec	Configuring Security for VPNs with IPsec
QoS options	*Cisco IOS Quality of Service Solutions Configuration Guide* on Cisco.com
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*
Security commands	*Cisco IOS Security Command Reference*
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standards	Title
No new or modified standards are supported by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec and Quality of Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 267: Feature Information for IPsec and Quality of Service

Feature Name	Releases	Feature Information
IPsec and Quality of Service	Cisco IOS XE Release 3.9S	The IPsec and Quality of Service feature allows Cisco IOS quality of service (QoS) policies to be applied to IP Security (IPsec) packet flows on the basis of a QoS group that can be added to the current Internet Security Association and Key Management Protocol (ISAKMP) profile. The following commands were introduced or modified: **qos-group.**

CHAPTER 202

VRF-Aware IPsec

The VRF-Aware IPsec feature introduces IP Security (IPsec) tunnel mapping to Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Restrictions for VRF-Aware IPsec

- If you are configuring the VRF-Aware IPsec feature using a crypto map configuration and the Inside VRF (IVRF) is not the same as the Front Door VRF (FVRF), this feature is not interoperable with unicast reverse path forwarding (uRPF) if uRPF is enabled on the crypto map interface. If your network requires uRPF, it is recommended that you use Virtual Tunnel Interface (VTI) for IPsec instead of crypto maps.
- The VRF-Aware IPsec feature does not allow IPsec tunnel mapping between VRFs. For example, it does not allow IPsec tunnel mapping from VRF vpn1 to VRF vpn2.
- When the VRF-Aware IPsec feature is used with a crypto map, this crypto map cannot use the global VRF as the IVRF and a non-global VRF as the FVRF. However, configurations based on virtual tunnel interfaces do not have that limitation. When VTIs or Dynamic VTIs (DVTIs) are used, the global VRF can be used as the IVRF together with a non-global VRF used as the FVRF.
- You must include the VRF in the **local-address** command when using the local address with VRF in the ISAKMP profile and keyring.

Information About VRF-Aware IPsec

VRF Instance

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and Cisco Express Forwarding (CEF) tables is maintained for each VPN customer.

MPLS Distribution Protocol

The MPLS distribution protocol is a high-performance packet-forwarding technology that integrates the performance and traffic management capabilities of data link layer switching with the scalability, flexibility, and performance of network-layer routing.

VRF-Aware IPsec Functional Overview

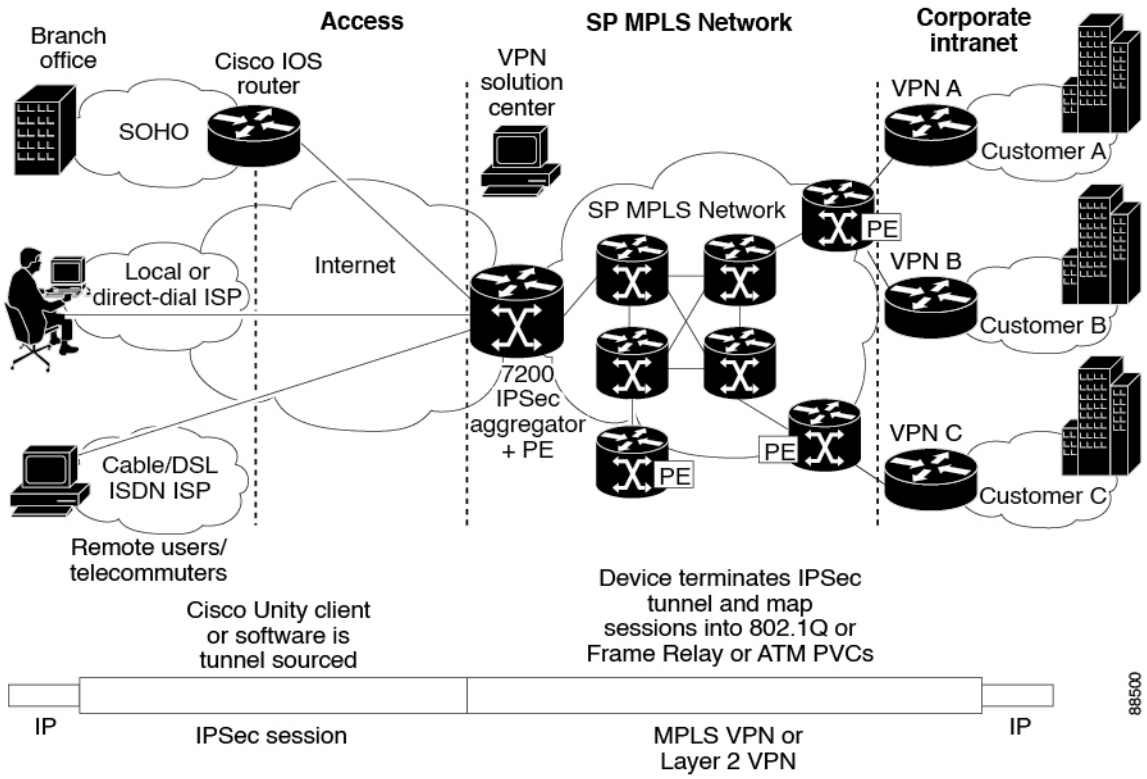
Front Door VRF (FVRF) and Inside VRF (IVRF) are central to understanding the feature.

Each IPsec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, which we shall call the FVRF, while the inner, protected IP packet belongs to another domain called the IVRF. Another way of stating the same thing is that the local endpoint of the IPsec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF.

One or more IPsec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

The diagram below is an illustration of a scenario showing IPsec to MPLS and Layer 2 VPNs.

Figure 101: IPsec to MPLS and Layer 2 VPNs



Packet Flow into the IPsec Tunnel

- A VPN packet arrives from the Service Provider MPLS backbone network to the PE and is routed through an interface facing the Internet.
- The packet is matched against the Security Policy Database (SPD), and the packet is IPsec encapsulated. The SPD includes the IVRF and the access control list (ACL).
- The IPsec encapsulated packet is then forwarded using the FVRF routing table.

Packet Flow from the IPsec Tunnel

- An IPsec-encapsulated packet arrives at the PE router from the remote IPsec endpoint.
- IPsec performs the Security Association (SA) lookup for the Security Parameter Index (SPI), destination, and protocol.
- The packet is decapsulated using the SA and is associated with IVRF.
- The packet is further forwarded using the IVRF routing table.

How to Configure VRF-Aware IPsec

Configuring Crypto Keyrings

A crypto keyring is a repository of preshared and Rivest, Shamir, and Adelman (RSA) public keys. There can be zero or more keyrings on the Cisco IOS router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
4. **description** *string*
5. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key*
6. **rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**]
7. **address** *ip-address*
8. **serial-number** *serial-number*
9. **key-string**
10. **text**
11. **quit**
12. **exit**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* [**vrf** *fvrf-name*] **Example:** `Router (config)# crypto keyring VPN1`	Defines a keyring with *keyring-name* as the name of the keyring and enters keyring configuration mode. • (Optional) The **vrf** keyword and *fvrf-name* argument imply that the keyring is bound to Front Door Virtual Routing and Forwarding (FVRF). The key in the keyring is searched if the local endpoint is in FVRF. If **vrf** is not specified, the keyring is bound to the global.

	Command or Action	Purpose
Step 4	**description** *string* **Example:** **Example:** `Router (config-keyring)# description The keys for VPN1`	(Optional) Specifies a one-line description of the keyring.
Step 5	**pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key* **Example:** `Router (config-keyring)# pre-shared-key address 10.72.23.11 key VPN1`	(Optional) Defines a preshared key by address or host name.
Step 6	**rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**] **Example:** `Router(config-keyring)# rsa-pubkey name host.vpn.com`	(Optional) Defines an RSA public key by address or host name and enters rsa-pubkey configuration mode. • The optional **encryption** keyword specifies that the key should be used for encryption. • The optional **signature** keyword specifies that the key should be used for signature. By default, the key is used for signature.
Step 7	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	(Optional) Defines the RSA public key IP address.
Step 8	**serial-number** *serial-number* **Example:** `Router(config-pubkey-key)# serial-number 1000000`	(Optional) Specifies the serial number of the public key. The value is from 0 through infinity.
Step 9	**key-string** **Example:** `Router (config-pubkey-key)# key-string`	Enters into the text mode in which you define the public key.
Step 10	**text** **Example:** `Router (config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973`	Specifies the public key. **Note** Only one public key may be added in this step.
Step 11	**quit** **Example:**	Quits to the public key configuration mode.

	Command or Action	Purpose
	`Router (config-pubkey)# quit`	
Step 12	**exit** **Example:** `Router (config-pubkey)# exit`	Exits to the keyring configuration mode.
Step 13	**exit** **Example:** `Router(config-keyring)# exit#`	Exits to global configuration mode.

Configuring ISAKMP Profiles

An ISAKMP profile is a repository for Internet Key Exchange (IKE) Phase 1 and IKE Phase 1.5 configuration for a set of peers. An ISAKMP profile defines items such as keepalive, trustpoints, peer identities, and XAUTH AAA list during the IKE Phase 1 and Phase 1.5 exchange. There can be zero or more ISAKMP profiles on the Cisco IOS router.

Note If traffic from the router to a certification authority (CA) (for authentication, enrollment, or for obtaining a certificate revocation list [CRL]) or to an Lightweight Directory Access Protocol (LDAP) server (for obtaining a CRL) needs to be routed via a VRF, the **vrf**command must be added to the trustpoint. Otherwise, the traffic uses the default routing table.

- If a profile does not specify one or more trustpoints, all trustpoints in the router will be used to attempt to validate the certificate of the peer (IKE main mode or signature authentication). If one or more trustpoints are specified, only those trustpoints will be used.

Note A router initiating IKE and a router responding to the IKE request should have symmetrical trustpoint configurations. For example, a responding router (in IKE Main Mode) performing RSA signature encryption and authentication might use trustpoints that were defined in the global configuration when sending the CERT-REQ payloads. However, the router might use a restricted list of trustpoints that were defined in the ISAKMP profile for the certificate verification. If the peer (the IKE initiator) is configured to use a certificate whose trustpoint is in the global list of the responding router but not in ISAKMP profile of the responding router, the certificate will be rejected. (However, if the initiating router does not know about the trustpoints in the global configuration of the responding router, the certificate can still be authenticated.)

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto isakmp profile** *profile-name*
4. **description** *string*
5. **vrf** *ivrf-name*
6. **keepalive** *seconds* **retry** *retry-seconds*
7. **self-identity** {**address** | **fqdn**| **user-fqdn** *user-fqdn*}
8. **keyring** *keyring-name*
9. **ca trust-point** {*trustpoint-name*}
10. **match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf*] | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}
11. **client configuration address** {**initiate** | **respond**}
12. **client authentication list** *list-name*
13. **isakmp authorization list** *list-name*
14. **initiate mode aggressive**
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an Internet Security Association and Key Management Protocol (ISAKMP) profile and enters into isakmp profile configuration mode.
Step 4	**description** *string* **Example:** `Router (conf-isa-prof)# description configuration for VPN profile`	(Optional) Specifies a one-line description of an ISAKMP profile.
Step 5	**vrf** *ivrf-name* **Example:** `Router (conf-isa-prof)# vrf VPN1`	(Optional) Maps the IPsec tunnel to a Virtual Routing and Forwarding (VRF) instance. **Note** The VRF also serves as a selector for matching the Security Policy Database (SPD). If the VRF is not specified in the ISAKMP profile, the IVRF of the IPsec tunnel will be the same as its FVRF.

	Command or Action	Purpose
Step 6	**keepalive** *seconds* **retry** *retry-seconds* **Example:** `Router (conf-isa-prof)# keepalive 60 retry 5`	(Optional) Allows the gateway to send dead peer detection (DPD) messages to the peer. • If not defined, the gateway uses the global configured value. • *seconds* --Number of seconds between DPD messages. The range is 10 to 3600 seconds. • **retry** *retry-seconds* --Number of seconds between retries if the DPD message fails. The range is 2 to 60 seconds.
Step 7	**self-identity** {**address** | **fqdn**| **user-fqdn** *user-fqdn*} **Example:** `Router (conf-isa-prof)# self-identity address`	(Optional) Specifies the identity that the local Internet Key Exchange (IKE) should use to identify itself to the remote peer. • If not defined, IKE uses the global configured value. • **address** --Uses the IP address of the egress interface. • **fqdn**-- Uses the fully qualified domain name (FQDN) of the router. • **user-fqdn** --Uses the specified value.
Step 8	**keyring** *keyring-name* **Example:** `Router (conf-isa-prof)# keyring VPN1`	(Optional) Specifies the keyring to use for Phase 1 authentication. • If the keyring is not specified, the global key definitions are used.
Step 9	**ca trust-point** {*trustpoint-name*} **Example:** `Router (conf-isa-prof)# ca trustpoint VPN1-trustpoint`	(Optional) Specifies a trustpoint to validate a Rivest, Shamir, and Adelman (RSA) certificate. • If no trustpoint is specified in the ISAKMP profile, all the trustpoints that are configured on the Cisco IOS router are used to validate the certificate.
Step 10	**match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf*] | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*} **Example:** `Router (conf-isa-prof)# match identity address 10.1.1.1`	Specifies the client IKE Identity (ID) that is to be matched. • **group** *group-name* --Matches the *group-name* with the ID type ID_KEY_ID. It also matches the *group-name* with the Organizational Unit (OU) field of the Distinguished Name (DN).· • **address** *address* [*mask*] *fvrf* --Matches the *address* with the ID type ID_IPV4_ADDR. The *mask* argument can be used to specify a range of addresses. The *fvrf* argument specifies that the address is in Front Door Virtual Routing and Forwarding (FVRF)

	Command or Action	Purpose
		• **host** *hostname* --Matches the *hostname* with the ID type ID_FQDN. • **host domain** *domain-name* --Matches the *domain-name* to the ID type ID_FQDN whose domain name is the same as the *domain-name*. Use this command to match all the hosts in the domain. • **user** *username* --Matches the *username* with the ID type ID_USER_FQDN· • **user domain** *domainname* --Matches the ID type ID_USER_FQDN whose domain name matches the *domainname*.
Step 11	**client configuration address** {**initiate** | **respond**} **Example:** `Router (conf-isa-prof)# client configuration address initiate`	(Optional) Specifies whether to initiate the mode configuration exchange or responds to mode configuration requests.
Step 12	**client authentication list** *list-name* **Example:** `Router (conf-isa-prof)# client authentication list xauthlist`	(Optional) AAA (authentication, authorization, and accounting) to use for authenticating the remote client during the extended authentication (XAUTH) exchange.
Step 13	**isakmp authorization list** *list-name* **Example:** `Router (conf-isa-prof)# isakmp authorization list ikessaaalist`	(Optional) Network authorization server for receiving the Phase 1 preshared key and other attribute-value (AV) pairs.
Step 14	**initiate mode aggressive** **Example:** `Router (conf-isa-prof)# initiate mode aggressive`	(Optional) Initiates aggressive mode exchange. • If not specified, IKE always initiates main mode exchange.
Step 15	**exit** **Example:** `Router (conf-isa-prof)# exit`	Exits to global configuration mode.

What to Do Next

Go to Configuring an ISAKMP Profile on a Crypto Map, on page 2608.

Configuring an ISAKMP Profile on a Crypto Map

An ISAKMP profile must be applied to the crypto map. The IVRF on the ISAKMP profile is used as a selector when matching the VPN traffic. If there is no IVRF on the ISAKMP profile, the IVRF will be equal to the FVRF. Perform this task to configure an ISAKMP profile on a crypto map.

Before you begin

Before configuring an ISAKMP profile on a crypto map, you must first configure your router for basic IPsec.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **isakmp-profile** *isakmp-profile-name*
4. **set isakmp-profile** *profile-name*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **isakmp-profile** *isakmp-profile-name* **Example:** `Router (config)# crypto map vpnmap isakmp-profile vpnprofile`	(Optional) Specifies the Internet Key Exchange and Key Management Protocol (ISAKMP) profile for the crypto map set and enters crypto map configuration mode. • The ISAKMP profile will be used during IKE exchange.
Step 4	**set isakmp-profile** *profile-name* **Example:** `Router (config-crypto-map)# set isakmp-profile vpnprofile`	(Optional) Specifies the ISAKMP profile to use when the traffic matches the crypto map entry.
Step 5	**exit** **Example:** `Router (config-crypto-map)# exit`	Exits to global configuration mode.

Configuring to Ignore Extended Authentication During IKE Phase 1 Negotiation

To ignore XAUTH during an IKE Phase 1 negotiation, use the **no crypto xauth** command. Use the **no crypto xauth** command if you do not require extended authentication for the Unity clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto xauth** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto xauth** *interface* **Example:** `Router(config)# no crypto xauth ethernet0`	Ignores XAUTH proposals for requests that are destined to the IP address of the interface. By default, Internet Key Exchange (IKE) processes XAUTH proposals.

Verifying VRF-Aware IPsec

To verify your VRF-Aware IPsec configurations, use the following **show** commands. These **show** commands allow you to list configuration information and security associations (SAs):

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa** [**map** *map-name*| **address** | **identity** | **interface** *interface* | **peer** [**vrf** *fvrf-name*] **address** | **vrf** *ivrf-name*] [**detail**]
3. **show crypto isakmp key**
4. **show crypto isakmp profile**
5. **show crypto key pubkey-chain rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name*| **address** | **identity** | **interface** *interface* | **peer** [**vrf** *fvrf-name*] **address** | **vrf** *ivrf-name*] [**detail**] **Example:** `Router# show crypto ipsec sa vrf vpn1`	Allows you to view the settings used by current security associations (SAs).
Step 3	**show crypto isakmp key** **Example:** `Router# show crypto isakmp key`	Lists all the keyrings and their preshared keys. • Use this command to verify your crypto keyring configuration.
Step 4	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all ISAKMP profiles and their configurations.
Step 5	**show crypto key pubkey-chain rsa** **Example:** `Router# show crypto key pubkey-chain rsa`	Views the RSA public keys of the peer that are stored on your router. • The output is extended to show the keyring to which the public key belongs.

Clearing Security Associations

The following **clear** commands allow you to clear SAs.

SUMMARY STEPS

1. **enable**
2. **clear crypto sa** [**counters** | **map** *map-name* | **peer**[**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**clear crypto sa** [**counters** | **map** *map-name* | **peer**[**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*] **Example:** `Router# clear crypto sa vrf VPN1`	Clears the IPsec security associations (SAs).

Troubleshooting VRF-Aware IPsec

To troubleshoot VRF-Aware IPsec, use the following **debug**commands:

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IP security (IPsec) events.
Step 3	**debug crypto isakmp** **Example:** `Router(config)# debug crypto isakmp`	Displays messages about Internet Key Exchange (IKE) events.

Debug Examples for VRF-Aware IPsec

The following sample debug outputs are for a VRF-aware IPsec configuration:

IPsec PE

```
Router# debug crypto ipsec
Crypto IPSEC debugging is on
IPSEC-PE#debug crypto isakmp
Crypto ISAKMP debugging is on
IPSEC-PE#debug crypto isakmp d
04:31:28: ISAKMP (0:12): purging SA., sa=6482B354, delme=6482B354
```

```
04:31:28: ISAKMP: Unlocking IKE struct 0x63C142F8 for declare_sa_dead(), count 0
IPSEC-PE#debug crypto isakmp detail
Crypto ISAKMP internals debugging is on
IPSEC-PE#
IPSEC-PE#
IPSEC-PE#
04:32:07: ISAKMP: Deleting peer node by peer_reap for 10.1.1.1: 63C142F8
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DC887D4E
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.68.1.1
04:32:55: ISAKMP cookie AA8F7B41 49A60E88
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DBC8E125
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 B4BDB5B7
04:32:55: ISAKMP (0:0): received packet from 10.1.1.1 dport 500 sport 500 Global (N) NEW
SA
04:32:55: ISAKMP: local port 500, remote port 500
04:32:55: ISAKMP: hash from 729FA94 for 619 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   B91E2C70 095A1346          9.,p.Z.F
64218CD0: 0EDB4CA6 8A46784F B314FD3B 00        .[L&.FxO.};.
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 F7ACF384
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 0C07C670
04:32:55: ISAKMP: insert sa successfully sa = 6482B354
04:32:55: ISAKMP (0:13): processing SA payload. message ID = 0
04:32:55: ISAKMP (0:13): processing ID payload. message ID = 0
04:32:55: ISAKMP (0:13): peer matches vpn2-ra profile
04:32:55: ISAKMP: Looking for a matching key for 10.1.1.1 in default
04:32:55: ISAKMP: Created a peer struct for 10.1.1.1, peer port 500
04:32:55: ISAKMP: Locking peer struct 0x640BBB18, IKE refcount 1 for
crypto_ikmp_config_initialize_sa
04:32:55: ISAKMP (0:13): Setting client config settings 648252B0
04:32:55: ISAKMP (0:13): (Re)Setting client xauth list and state
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13) Authentication by xauth preshared
04:32:55: ISAKMP (0:13): Checking ISAKMP transform 1 against priority 1 policy
04:32:55: ISAKMP:      encryption AES-CBC
04:32:55: ISAKMP:      hash SHA
04:32:55: ISAKMP:      default group 14
04:32:55: ISAKMP:      auth XAUTHInitPreShared
04:32:55: ISAKMP:      life type in seconds
04:32:55: ISAKMP:      life duration (VPI) of 0x0 0x20 0xC4 0x9B
04:32:55: ISAKMP (0:13): atts are acceptable. Next payload is 3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13): processing KE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing NONCE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is DPD
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 175 mismatch
04:32:55: ISAKMP (0:13): vendor ID is XAUTH
```

```
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): claimed IOS but failed authentication
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is Unity
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_READY  New State = IKE_R_AM_AAA_AWAIT
04:32:55: ISAKMP cookie gen for src 11.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 7AE6E1DF
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 4 AA 31 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP (0:13): SKEYID state generated
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D70:                           0D000014              ....
63E66D80: 12F5F28C 457168A9 702D9FE2 74CC0100  .ur.Eqh)p-.btL..
63E66D90: 00                                   .
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D90: 0D000014 AFCAD713 68A1F1C9 6B8696FC  ..../JW.h!qIk..|
63E66DA0: 77570100 00                         wW...
04:32:55: ISAKMP (0:13): constructed NAT-T vendor-03 ID
04:32:55: ISAKMP (0:13): SA is doing pre-shared key authentication plus XAUTH using id type
 ID_IPV4_ADDR
04:32:55: ISAKMP (13): ID payload
        next-payload : 10
        type         : 1
        addr         : 172.16.1.1
        protocol     : 17
        port         : 0
        length       : 8
04:32:55: ISAKMP (13): Total payload length: 12
04:32:55: ISAKMP (0:13): constructed HIS NAT-D
04:32:55: ISAKMP (0:13): constructed MINE NAT-D
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) AG_INIT_EXCH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, PRESHARED_KEY_REPLY
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM_AAA_AWAIT  New State = IKE_R_AM2
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B D99DA70D
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 9C69F917
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 00583224
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 C1B006EE
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
AG_INIT_EXCH
04:32:55: ISAKMP: hash from 7003A34 for 132 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   D1202D99 2BB49D38          Q -.+4.8
64218CD0: B8FBB1BE 7CDC67D7 4E26126C 63        8{1>|\gWN&.lc
04:32:55: ISAKMP (0:13): processing HASH payload. message ID = 0
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc my hash for NAT-D
```

```
04:32:55: ISAKMP (0:13): NAT match MINE hash
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc his hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match HIS hash
04:32:55: ISAKMP (0:13): processing NOTIFY INITIAL_CONTACT protocol 1
        spi 0, message ID = 0, sa = 6482B354
04:32:55: ISAKMP (0:13): Process initial contact,
bring down existing phase 1 and 2 SA's with local 172.16.1.1 remote 10.1.1.1 remote port
500
04:32:55: ISAKMP (0:13): returning IP addr to the address pool
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 05D315C5
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 041A85A6
04:32:55: ISAKMP (0:13): SA has been authenticated with 10.1.1.1
04:32:55: ISAKMP: Trying to insert a peer 172.16.1.1/10.1.1.1/500/,  and inserted
successfully.
04:32:55: ISAKMP: set new node -803402627 to CONF_XAUTH
04:32:55: IPSEC(key_engine): got a queue event...
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE

04:32:55: ISAKMP (0:13): purging node -803402627
04:32:55: ISAKMP: Sending phase 1 responder lifetime 86400
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM2  New State = IKE_P1_COMPLETE
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.168.1.1
04:32:55: ISAKMP cookie AA8F7B41 25EEF256
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): Need XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:32:55: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_XAUTH_AAA_START_LOGIN_AWAIT
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 2CCFA491
04:32:55: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP: set new node -1447732198 to CONF_XAUTH
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_NAME_V2
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD_V2
04:32:55: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = -1447732198
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) CONF_XAUTH

04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_START_LOGIN
04:32:55: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_START_LOGIN_AWAIT  New State =
IKE_XAUTH_REQ_SENT
04:33:00: ISAKMP (0:13): retransmitting phase 2 CONF_XAUTH    -1447732198 ...
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): retransmitting phase 2 -1447732198 CONF_XAUTH
04:33:00: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) CONF_XAUTH

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 124D4618
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B0C91917
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
```

```
04:33:03: ISAKMP cookie 3123100B 0E294692
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 091A7695
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292D74 for 92 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   84A1AF24 5D92B116          .!/$].1.
64218CD0: FC2C6252 A472C5F8 152AC860 63        |,bR$rEx.*H`c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
-1447732198
04:33:03: ISAKMP: Config payload REPLY
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_NAME_V2
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_PASSWORD_V2
04:33:03: ISAKMP (0:13): deleting node -1447732198 error FALSE reason "done with xauth
request/reply exchange"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REPLY
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_REQ_SENT  New State =
IKE_XAUTH_AAA_CONT_LOGIN_AWAIT
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 A1B3E684
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP: set new node 524716665 to CONF_XAUTH
04:33:03: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = 524716665
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) CONF_XAUTH

04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_CONT_LOGIN
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_CONT_LOGIN_AWAIT  New State =
IKE_XAUTH_SET_SENT
004:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 5C83A09D
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 2BEBEFD4
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B DA00A46B
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 FDD27773
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292A34 for 68 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   5034B99E B8BA531F          P49.8:S.
64218CD0: 6267B8BD F3006989 DC118796 63        bg8=s.i.\...c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID = 524716665
04:33:03: ISAKMP: Config payload ACK
04:33:03: ISAKMP (0:13):        XAUTH ACK Processed
04:33:03: ISAKMP (0:13): deleting node 524716665 error FALSE reason "done with transaction"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_ACK
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_SET_SENT  New State = IKE_P1_COMPLETE
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 E0BB50E9
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
```

```
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 7794EF6E
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 C035AAE5
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B F1FCC25A
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 31744F44
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R) QM_IDLE

04:33:03: ISAKMP: set new node -1639992295 to QM_IDLE
04:33:03: ISAKMP: hash from 7293A74 for 100 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   9D7DF4DF FE3A6403          .}t_~:d.
64218CD0: 3F1D1C59 C5D138CE 50289B79 07        ?..YEQ8NP(.y.
04:33:03: ISAKMP (0:13): processing transaction payload from 10.1.1.1. message ID =
-1639992295
04:33:03: ISAKMP: Config payload REQUEST
04:33:03: ISAKMP (0:13): checking request:
04:33:03: ISAKMP:    IP4_ADDRESS
04:33:03: ISAKMP:    IP4_NETMASK
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    SPLIT_INCLUDE
04:33:03: ISAKMP:    DEFAULT_DOMAIN
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REQUEST
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State = IKE_CONFIG_AUTHOR_AAA_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B02E0D67
04:33:03: ISAKMP:        isadb_post_process_list: crawler: C 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP (0:13): attributes sent in message:
04:33:03:         Address: 10.2.0.0
04:33:03: ISAKMP (0:13): allocating address 10.4.1.4
04:33:03: ISAKMP: Sending private address: 10.4.1.4
04:33:03: ISAKMP: Sending DEFAULT_DOMAIN default domain name: vpn2.com
04:33:03: ISAKMP (0:13): responding to peer config from 10.1.1.1. ID = -1639992295
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) CONF_ADDR

04:33:03: ISAKMP (0:13): deleting node -1639992295 error FALSE reason ""
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_GROUP_ATTR
04:33:03: ISAKMP (0:13): Old State = IKE_CONFIG_AUTHOR_AAA_AWAIT  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 881D5411
04:33:03: ISAKMP cookie gen for src 11.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 6FD82541
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
```

```
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 8A94C1BE
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 F3BA766D
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R) QM_IDLE

04:33:03: ISAKMP: set new node 17011691 to QM_IDLE
04:33:03: ISAKMP: hash from 70029F4 for 540 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   AFBA30B2 55F5BC2D          /:02Uu<-
64218CD0: 3A86B1C9 00D2F5BA 77BF5589 07        :.1I.Ru:w?U..
04:33:03: ISAKMP (0:13): processing HASH payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing SA payload. message ID = 17011691
04:33:03: ISAKMP (0:13): Checking IPSec proposal 1
04:33:03: ISAKMP: transform 1, ESP_AES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-SHA
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: IPSEC(validate_transform_proposal): transform proposal not supported for identity:

    {esp-aes esp-sha-hmac}
04:33:03: ISAKMP (0:13): IPSec policy invalidated proposal
04:33:03: ISAKMP (0:13): Checking IPSec proposal 2
04:33:03: ISAKMP: transform 1, ESP_AES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-SHA
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: ISAKMP (0:13): processing NONCE payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): asking for 1 spis from ipsec
04:33:03: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
```

```
04:33:03: ISAKMP (0:13): Old State = IKE_QM_READY  New State = IKE_QM_SPI_STARVE
04:33:03: IPSEC(key_engine): got a queue event...
04:33:03: IPSEC(spi_response): getting spi 2749516541 for SA
        from 172.18.1.1     to 10.1.1.1        for prot 3
04:33:03: ISAKMP: received ke message (2/1)
04:33:04: ISAKMP (13): ID payload
        next-payload : 5
        type         : 1
        addr         : 10.4.1.4
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (13): ID payload
        next-payload : 11
        type         : 4
        addr         : 0.0.0.0
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE

04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_IPSEC, IKE_SPI_REPLY
04:33:04: ISAKMP (0:13): Old State = IKE_QM_SPI_STARVE  New State = IKE_QM_R_QM2
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B 93DE46D2
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 088A0A16
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B A8F23F73
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 93D8D879
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R) QM_IDLE

04:33:04: ISAKMP: hash from 7290DB4 for 60 bytes
04:33:04: ISAKMP: Packet hash:
64218CC0:                   4BB45A92 7181A2F8          K4Z.q."x
64218CD0: 73CC12F8 091875C0 054F77CD 63        sL.x..u@.OwMc
04:33:04: ISAKMP: Locking peer struct 0x640BBB18, IPSEC refcount 1 for stuff_ke
04:33:04: ISAKMP (0:13): Creating IPSec SAs
04:33:04:         inbound SA from 10.1.1.1 to 172.18.1.1 (f/i)  0/ 2
        (proxy 10.4.1.4 to 0.0.0.0)
04:33:04:         has spi 0xA3E24AFD and conn_id 5127 and flags 2
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04:         outbound SA from 172.18.1.1     to 10.1.1.1        (f/i)  0/ 2 (proxy
0.0.0.0         to 10.4.1.4       )
04:33:04:         has spi 1343294712 and conn_id 5128 and flags A
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04: ISAKMP (0:13): deleting node 17011691 error FALSE reason "quick mode done (await)"
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:04: ISAKMP (0:13): Old State = IKE_QM_R_QM2  New State = IKE_QM_PHASE2_COMPLETE
04:33:04: IPSEC(key_engine): got a queue event...
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac ,
```

```
    lifedur= 2147483s and 4608000kb,
    spi= 0xA3E24AFD(2749516541), conn_id= 5127, keysize= 0, flags= 0x2
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-aes esp-sha-hmac,
    lifedur= 2147483s and 4608000kb,
    spi= 0x50110CF8(1343294712), conn_id= 5128, keysize= 0, flags= 0xA
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:04: IPSEC(rte_mgr): VPN Route Added 10.4.1.4 255.255.255.255 via 10.1.1.1 in vpn2
04:33:04: IPSEC(add mtree): src 0.0.0.0, dest 10.4.1.4, dest_port 0
04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.18.1.1, sa_prot= 50,
    sa_spi= 0xA3E24AFD(2749516541),
    sa_trans= esp-aes esp-sha-hmac, sa_conn_id= 5127
04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0x50110CF8(1343294712),
    sa_trans= esp-aes esp-sha-hmac, sa_conn_id= 5128
04:33:53: ISAKMP (0:13): purging node -1639992295
04:33:54: ISAKMP (0:13): purging node 17011691
```

Configuration Examples for VRF-Aware IPsec

Example Static IPsec-to-MPLS VPN

The following sample shows a static configuration that maps IPsec tunnels to MPLS VPNs. The configurations map IPsec tunnels to MPLS VPNs “VPN1” and “VPN2.” Both of the IPsec tunnels terminate on a single public-facing interface.

IPsec PE Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto keyring vpn1
  pre-shared-key address 172.16.1.1 key vpn1
crypto keyring vpn2
  pre-shared-key address 10.1.1.1 key vpn2
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
```

```
!
crypto isakmp profile vpn2
 vrf vpn2
 keyring vpn2
 match identity address 10.1.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
crypto ipsec transform-set vpn2 esp-aes esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
crypto map crypmap 3 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set vpn2
 set isakmp-profile vpn2
 match address 102
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route 10.1.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
ip route vrf vpn2 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
access-list 102 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPsec Customer Provided Edge (CPE) Configuration for VPN1

```
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
crypto isakmp key vpn1 address 172.18.1.1
!
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPsec CPE Configuration for VPN2

```
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
!
crypto isakmp key vpn2 address 172.18.1.1
!
!
crypto ipsec transform-set vpn2 esp-aes esp-sha-hmac
!
crypto map vpn2 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn2
 match address 101
!
interface FastEthernet0
 ip address 10.1.1.1 255.255.255.0
 crypto map vpn2
!
interface FastEthernet1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

Example IPsec-to-MPLS VPN Using RSA Encryption

The following example shows an IPsec-to-MPLS configuration using RSA encryption:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto isakmp policy 10
 authentication rsa-encr
!
crypto keyring vpn1
 rsa-pubkey address 172.16.1.1 encryption
  key-string
   305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DBF381 00DDECC8
   DC4AA490 40320C52 9912D876 EB36717C 63DCA95C 7E5EC02A 84F276CE 292B42D7
   D664F324 3726F4E0 39D33093 ECB81B95 482511A5 F064C4B3 D5020301 0001
   quit
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
```

```
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPsec CPE Configuration for VPN1

```
crypto isakmp policy 10
 authentication rsa-encr
!
crypto key pubkey-chain rsa
 addressed-key 172.18.1.1 encryption
 key-string
  3082011B 300D0609 2A864886 F70D0101 01050003 82010800 30820103 0281FB00
  C90CC78A 6002BDBA 24683396 B7D7877C 16D08C47 E00C3C10 63CF13BC 4E09EA23
  92EB8A48 4113F5A4 8796C8BE AD7E2DC1 3B0742B6 7118CE7C 1B0E21D1 AA9724A4
  4D74FCEA 562FF225 A2B11F18 E53C4415 61C3B741 3A06E75D B4F9102D 6163EE40
  16C68FD7 6532F660 97B59118 9C8DE3E5 4E2F2925 BBB87FCB 95223D4E A5E362DB
  215CB35C 260080805 17BBE1EF C3050E13 031F3D5B 5C22D16C FC8B1EC5 074F07A5
  D050EC80 7890D9C5 EC20D6F0 173FE2BA 89F5B5F9 2EADC9A6 D461921E 3D5B60016
  ABB8B6B9 E2124A21 93F0E4AE B487461B E7F1F1C4 032A0B0E 80DC3E15 CB268EC9
  5D76B9BD 3C78CB75 CE9F68C6 484D6573 CBC3EB59 4B5F3999 8F9D0203 010001
  quit
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

Example IPsec-to-MPLS VPN with RSA Signatures

The following shows an IPsec-to-MPLS VPN configuration using RSA signatures:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
```

```
!
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
!
crypto ca certificate chain bombo
 certificate 03C0
 308203BF 308202A7 A0030201 02020203 C0300D06 092A8648 86F70D01 01050500
 . . .
 quit
 certificate ca 01
 30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
 . . .
 quit
!
crypto isakmp profile vpn1
 vrf vpn1
 ca trust-point bombo
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.31.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
!
```

IPsec CPE Configuration for VPN1

```
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
 !
crypto ca certificate chain bombo
 certificate 03BF
  308203BD 308202A5 A0030201 02020203 BF300D06 092A8648 86F70D01 01050500
  . . .
  quit
 certificate ca 01
  30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  . . .
  quit
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
```

```
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

Example IPsec Remote Access-to-MPLS VPN

The following shows an IPsec remote access-to-MPLS VPN configuration. The configuration maps IPsec tunnels to MPLS VPNs. The IPsec tunnels terminate on a single public-facing interface.

PE Router Configuration

```
aaa new-model
!
aaa group server radius vpn1
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn1
!
aaa group server radius vpn2
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn2
!
aaa authorization network aaa-list group radius
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto isakmp profile vpn1-ra
   vrf vpn1
   match identity group vpn1-ra
   client authentication list vpn1
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
crypto isakmp profile vpn2-ra
   vrf vpn2
   match identity group vpn2-ra
   client authentication list vpn2
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
!
!
crypto ipsec transform-set vpn1 esp-aes esp-sha-hmac
crypto ipsec transform-set vpn2 esp-aes esp-sha-hmac
!
crypto dynamic-map vpn1 1
 set transform-set vpn1
 set isakmp-profile vpn1-ra
```

```
 reverse-route
!
crypto dynamic-map vpn2 1
 set transform-set vpn2
 set isakmp-profile vpn2-ra
 reverse-route
!
!
crypto map ra 1 ipsec-isakmp dynamic vpn1
crypto map ra 2 ipsec-isakmp dynamic vpn2
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map ra
!
ip local pool vpn1-ra 10.4.1.1 10.4.1.254 group vpn1-ra
ip local pool vpn2-ra 10.4.1.1 10.4.1.254 group vpn2-ra
!
```

Upgrade from Previous Versions of the Cisco Network-Based IPsec VPN Solution

The VRF-Aware IPsec feature in the Cisco network-based IPsec VPN solution release 1.5 requires that you change your existing configurations. The following sample configurations indicate the changes you must make to your existing configurations.

Site-to-Site Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site configuration upgrade from a previous version of the network-based IPsec VPN solution to the Cisco network-based IPsec VPN solution release 1.5:

Previous Version Site-to-Site Configuration

```
crypto isakmp key VPN1 address 172.21.25.74
 crypto isakmp key VPN2 address 172.21.21.74
!
 crypto ipsec transform-set VPN1 esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
```

```
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site Configuration

The following is an upgraded version of the same site-to-site configuration to the Cisco network-based IPsec VPN solution release 1.5 solution:

Note You must change two keyrings. The VRF-Aware Upset feature requires that keys be associated with a VRF if the IKE local endpoint is in the VRF.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
  pre-shared-key address 172.21.21.74 key VPN2
!
 crypto ipsec transform-set VPN1 esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
  set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
  set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a remote access configuration upgrade from a previous version of the network-based IPsec VPN solution to the Cisco network-based IPsec VPN solution release 1.5:

Previous Version Remote Access Configuration

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
```

```
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-aes esp-sha-hmac
!
 crypto dynamic-map VPN1-RA 1
  set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
  reverse-route
!
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Remote Access Configuration

In the following instance, there is no upgrade; it is recommended that you change to the following configuration:

```
crypto isakmp client configuration group VPN1-RA-GROUP
  key VPN1-RA
  pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
  client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
```

```
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-aes esp-sha-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Combination Site-to-Site and Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site and remote access configuration upgrade from a previous version of the network-based IPsec VPN solution to the Cisco network-based IPsec VPN solution release 1.5:

Previous Version Site-to-Site and Remote Access Configuration

```
crypto isakmp key VPN1 address 172.21.25.74 no-xauth
crypto isakmp key VPN2 address 172.21.21.74 no-xauth
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1 esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-aes esp-sha-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
```

```
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site and Remote Access Configuration

You must upgrade to this configuration:

Note For site-to-site configurations that do not require XAUTH, configure an ISAKMP profile without XAUTH configuration. For remote access configurations that require XAUTH, configure an ISAKMP profile with XAUTH.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
```

```
 crypto isakmp profile VPN1
 keyring VPN1-KEYS
 match identity address 172.21.25.74 VPN1
!
 crypto isakmp profile VPN2
 keyring VPN2-KEYS
 match identity address 172.21.21.74 VPN2
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1 esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-aes esp-sha-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 set isakmp-profile VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 set isakmp-profile VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Additional References

Related Documents

Related Topic	Document Title
IPsec configuration tasks	" Configuring Security for VPNs with IPsec"
IPsec commands	*Cisco IOS Security Command Reference*
IKE Phase 1 and Phase 2, aggressive mode, and main mode	"Configuring Internet Key Exchange for IPsec VPNs"
IKE dead peer detection	"Easy VPN Server"
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRF-Aware IPsec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 268: Feature Information for VRF-Aware IPsec

Feature Name	Releases	Feature Information
VRF-Aware IPsec	12.2(15)T	The VRF-Aware IPsec feature introduces IP Security (IPsec) tunnel mapping to Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using the VRF-Aware IPsec feature, you can map IPsec tunnels to Virtual Routing and Forwarding (VRF) instances using a single public-facing address. This feature was introduced in Cisco IOS Release 12.2(15)T. The following sections provide information about this feature: The following commands were introduced or modified: **address, ca trust-point, client authentication list, client configuration address, crypto isakmp profile, crypto keyring, crypto map isakmp-profile**, **initiate-mode, isakmp authorization list, keepalive (isakmp profile), keyring, key-string, match identity, no crypto xauth, pre-shared-key, quit, rsa-pubkey, self-identity, serial-number, set isakmp-profile, show crypto isakmp key, show crypto isakmp profile, vrf, clear crypto sa, crypto isakmp peer, crypto map isakmp-profile, show crypto dynamic-map, show crypto ipsec sa, show crypto isakmp sa, show crypto map (IPsec)**.
	15.1(1)S	This feature was integrated into Cisco IOS Release 15.1(1)S.

Glossary

CA --certification authority. CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CLI --command-line-interface. CLI is an interface that allows the user to interact with the operating system by entering commands and optional arguments. The UNIX operating system and DOS provide CLIs.

client --Corresponding IPsec IOS peer of the UUT in the Multi Protocol Label Switching (MPLS) network.

dead peer --IKE peer that is no longer reachable.

DN --Distinguished Name. A DN is the global, authoritative name of an entry in the Open System Interconnection (OSI Directory [X.500]).

FQDN --fully qualified domain name. A FQDN is the full name of a system rather than just its host name. For example, aldebaran is a host name, and aldebaran.interop.com is an FQDN.

FR --Frame Relay. FR is an industry-standard, switch-data-link-layer protocol that handles multiple virtual circuits using high-level data link (HDLC) encapsulation between connected devices. Frame Relay is more efficient than X.25, the protocol for which it generally is considered a replacement.

FVRF --Front Door Virtual Routing and Forwarding (VRF) repository. FVRF is the VRF used to route the encrypted packets to the peer.

IDB --Interface descriptor block. An IDB subblock is an area of memory that is private to an application. This area stores private information and states variables that an application wants to associate with an IDB or an interface. The application uses the IDB to register a pointer to its subblock, not to the contents of the subblock itself.

IKE --Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPsec) that require keys. Before any IPsec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IKE keepalive --Bidirectional mechanism for determining the liveliness of an IKE peer.

IPsec --Security protocol for IP.

IVRF --Inside Virtual Routing and Forwarding. IVRF is the VRF of the plaintext packets.

MPLS --Multiprotocol Label Switching. MPLS is a switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

RSA --Rivest, Shamir, and Adelman are the inventors of the RSA technique. The RSA technique is a public-key cryptographic system that can be used for encryption and authentication.

SA --Security Association. SA is an instance of security policy and keying material applied to a data flow.

VPN --Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP or IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

VRF --Virtual Route Forwarding. VRF is A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

XAUTH --Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

CHAPTER 203

IKE Initiate Aggressive Mode

The IKE: Initiate Aggressive Mode feature allows you to specify RADIUS tunnel attributes for an IP security (IPsec) peer and to initiate an Internet Key Exchange (IKE) aggressive mode negotiation with the tunnel attributes. This feature is best implemented in a crypto hub-and-spoke scenario, by which the spokes initiate IKE aggressive mode negotiation with the hub by using the preshared keys that are specified as tunnel attributes and stored on the AAA server. This scenario is scalable because the preshared keys are kept at a central repository (the AAA server) and more than one hub router and one application can use the information.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for IKE Initiate Aggressive Mode

Before configuring the Initiate Aggressive Mode IKE feature, you must perform the following tasks:

- Configure AAA
- Configure an IPsec Transform
- Configure a static crypto map
- Configure an Internet Security Association and Key Management Protocol (ISAKMP) policy
- Configure a dynamic crypto map

Restrictions for IKE Initiate Aggressive Mode

TED Restriction

This feature is not intended to be used with a dynamic crypto map that uses Tunnel Endpoint Discovery (TED) to initiate tunnel setup. TED is useful in configuring a full mesh setup, which requires an AAA server at each site to store the preshared keys for the peers; this configuration is not practical for use with this feature.

Tunnel-Client-Endpoint ID Types

Only the following ID types can be used in this feature:

- ID_IPV4 (IPV4 address)
- ID_FQDN (fully qualified domain name, for example “foo.cisco.com”)
- ID_USER_FQDN (e-mail address)

Information About IKE Initiate Aggressive Mode

Overview

The IKE: Initiate Aggressive Mode feature allows you to configure IKE preshared keys as RADIUS tunnel attributes for IPsec peers. Thus, you can scale your IKE preshared keys in a hub-and-spoke topology.

Although IKE preshared keys are simple to understand and easy to deploy, they do not scale well with an increasing number of users and are therefore prone to security threats. Instead of keeping your preshared keys on the hub router, this feature allows you to scale your preshared keys by storing and retrieving them from an authentication, authorization, and accounting (AAA) server. The preshared keys are stored in the AAA server as Internet Engineering Task Force (IETF) RADIUS tunnel attributes and are retrieved when a user tries to “speak” to the hub router. The hub router retrieves the preshared key from the AAA server and the spokes (the users) initiate aggressive mode to the hub by using the preshared key that is specified in the Internet Security Association Key Management Policy (ISAKMP) peer policy as a RADIUS tunnel attribute.

RADIUS Tunnel Attributes

To initiate an IKE aggressive mode negotiation, the Tunnel-Client-Endpoint (66) and Tunnel-Password (69) attributes must be configured in the ISAKMP peer policy. The Tunnel-Client-Endpoint attribute will be communicated to the server by encoding it in the appropriate IKE identity payload; the Tunnel-Password attribute will be used as the IKE preshared key for the aggressive mode negotiation

How to Configure IKE Initiate Aggressive Mode

Configuring RADIUS Tunnel Attributes

To configure the Tunnel-Client-Endpoint and Tunnel-Password attributes within the ISAKMP peer configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **isakmp authorization list** *list-name*
4. **crypto isakmp peer** {**ip-address** *ip-address* | **fqdn** *fqdn*}
5. **set aggressive-mode client-endpoint** *client-endpoint*
6. **set aggressive-mode password** *password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **isakmp authorization list** *list-name* **Example:** `Router (config)# crypto map testmap10 isakmp authorization list list ike`	Enables IKE querying of AAA for tunnel attributes in aggressive mode.
Step 4	**crypto isakmp peer** {**ip-address** *ip-address* | **fqdn** *fqdn*} **Example:** `Router (config)# crypto isakmp peer ip address 10.10.10.1`	Enables an IPsec peer for IKE querying of AAA for tunnel attributes in aggressive mode and enters ISAKMP policy configuration mode.
Step 5	**set aggressive-mode client-endpoint** *client-endpoint* **Example:** `Router (config-isakmp)# set aggressive-mode client-endpoint user-fqdn user@cisco.com`	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.

	Command or Action	Purpose
Step 6	**set aggressive-mode password** *password* **Example:** `Router (config-isakmp)#set aggressive-mode password cisco123`	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

Verifying RADIUS Tunnel Attribute Configurations

To verify that the Tunnel-Client-Endpoint and Tunnel-Password attributes have been configured within the ISAKMP peer policy, use the **show running-config**global configuration command.

Troubleshooting Tips

To troubleshoot the IKE: Initiate Aggressive Mode feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug aaa authorization**
3. **debug crypto isakmp**
4. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information about AAA authorization.
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.
Step 4	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.

Configuration Examples for IKE Initiate Aggressive Mode

Hub Configuration Example

The following example shows how to configure a hub for a hub-and-spoke topology that supports aggressive mode using RADIUS tunnel attributes:

```
!The AAA configurations are as follows:
aaa new-model
aaa authorization network ike group radius
aaa authentication login default group radius
!
! The Radius configurations are as follows:
radius-server host 10.1.1.1 auth-port 1645 acct-port 1646
radius-server key rad123
!
! The IKE configurations are as follows:
crypto isakmp policy 1
 authentication pre-share
!
! The IPsec configurations are as follows:
crypto ipsec transform-set trans1 esp-aes esp-sha-hmac
!
crypto dynamic-map Dmap 10
 set transform-set trans1
!
crypto map Testtag isakmp authorization list ike
crypto map Testtag 10 ipsec-isakmp dynamic Dmap
!
interface FastEthernet0
 ip address 10.4.4.1 255.255.255.0
 crypto map Testtag
!
interface FastEthernet1
 ip address 10.2.2.1 255.255.255.0
```

Spoke Configuration Example

The following example shows how to configure a spoke for a hub-and-spoke topology that supports aggressive mode using RADIUS tunnel attributes:

```
!The IKE configurations are as follows:
crypto isakmp policy 1
 authentication pre-share
!
! The IPsec configurations are as follows:
crypto ipsec transform-set trans1 esp-aes esp-sha-hmac
 access-list 101 permit ip 10.3.3.0 0.0.0.255 10.2.2.0 0.0.0.255
!
! Initiate aggressive mode using Radius tunnel attributes
crypto isakmp peer address 10.4.4.1
 set aggressive-mode client-endpoint user-fqdn user@cisco.com
 set aggressive-mode password cisco123
!
crypto map Testtag 10 ipsec-isakmp
 set peer 10.4.4.1
```

```
 set transform-set trans1
 match address 101
!
interface FastEthernet0
 ip address 10.5.5.1 255.255.255.0
 crypto map Testtag
!
interface FastEthernet1
 ip address 10.3.3.1 255.255.255.0
```

RADIUS User Profile Example

The following is an example of a user profile on a RADIUS server that supports the Tunnel-Client-Endpoint and Tunnel-Password attributes:

```
user@cisco.com  Password = "cisco", Service-Type = Outbound
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Type = :1:ESP,
    Cisco:Avpair = "ipsec:tunnel-password=cisco123",
    Cisco:Avpair = "ipsec:key-exchange=ike"
```

Additional References

The following sections provide references related to the IKE: Initiate Aggressive Mode feature.

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*
Configuring authentication	Configuring Authentication
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
• RFC 2409 • RFC 2868	• RFC 2409, *The Internet Key Exchange* • RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IKE Initiate Aggressive Mode

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 269: Feature Information for IKE: Initiate Aggressive Mode

Feature Name	Releases	Feature Information
IKE: Initiate Aggressive Mode	Cisco IOS XE Release 2.1	The IKE: Initiate Aggressive Mode feature allows you to specify RADIUS tunnel attributes for an IPsec peer and to initiate an IKE aggressive mode negotiation with the tunnel attributes. The following commands were introduced or modified: **crypto isakmp peer**, **set aggressive-mode client-endpoint**, **set aggressive-mode password**.

PART XXI

FlexVPN and Internet Key Exchange

CHAPTER 204

Introduction to FlexVPN

Internet Key Exchange Version 2 (IKEv2), a next-generation key management protocol based on RFC 4306, is an enhancement of the IKE Protocol. IKEv2 is used for performing mutual authentication and establishing and maintaining security associations (SAs).

FlexVPN is Cisco's implementation of the IKEv2 standard featuring a unified paradigm and CLI that combines site to site, remote access, hub and spoke topologies and partial meshes (spoke to spoke direct). FlexVPN offers a simple but modular framework that extensively uses the tunnel interface paradigm while remaining compatible with legacy VPN implementations using crypto maps.

This guide contains the following modules:

Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Remote Access

This module describes IKEv2 CLI and is divided into basic and advanced sections.

The basic section introduces basic IKEv2 commands and describes IKEv2 smart defaults and the mandatory IKEv2 commands required for FlexVPN remote access. This module is a prerequisite for understanding subsequent chapters.

The advanced section describes global IKEv2 commands and how to override the default IKEv2 commands.

Configuring FlexVPN Server

This module describes FlexVPN server features, IKEv2 commands required to configure FlexVPN server, remote access clients and the supported RADIUS attributes.

Configuring FlexVPN Client

This module describes FlexVPN client features and the IKEv2 commands required for FlexVPN client.

Configuring IKEv2 Load Balancer

This module describes the IKEv2 Load Balancer Support feature and the IKEv2 commands required to configure the IKEv2 Load Balancer.

Configuring IKEv2 Fragmentation

The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document.

Configuring IKEv2 Reconnect

The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client.

Configuring IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices.

Configuring IKEv2 Change of Authorization Support

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session.

Configuring Aggregate Authentication

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP

authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server.

Appendix: FlexVPN RADIUS Attributes

This module describes the RADIUS attributes supported by FlexVPN server.

Appendix: IKEv2 and Legacy VPNs

This module contains configuration examples on how to configure legacy VPNs such as crypto maps and DMVPN with Internet Key Exchange Version 2 (IKEv2).

CHAPTER 205

Configuring Internet Key Exchange Version 2

This module contains information about and instructions for configuring basic and advanced Internet Key Exchange Version 2 (IKEv2). The tasks and configuration examples for IKEv2 in this module are divided as follows:

- Basic IKEv2—Provides information about basic IKEv2 commands, IKEv2 smart defaults, basic IKEv2 profile, and IKEv2 key ring.
- Advanced IKEv2—Provides information about global IKEv2 commands and how to override IKEv2 smart defaults.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Configuring Internet Key Exchange Version 2

You should be familiar with the concepts and tasks described in the "Configuring Security for VPNs with IPsec" module.

Restrictions for Configuring Internet Key Exchange Version 2

You cannot configure an option that is not supported on a specific platform. For example, in a security protocol, the capability of the hardware-crypto engine is important, and you cannot specify the Triple Data Encryption

Standard (3DES) or the Advanced Encryption Standard (AES) type of encryption transform in a nonexportable image, or specify an encryption algorithm that a crypto engine does not support.

Information About Internet Key Exchange Version 2

IKEv2 Supported Standards

Cisco implements the IP Security (IPsec) Protocol standard for use in Internet Key Exchange Version 2 (IKEv2).

Note Cisco no longer recommends using DES or MD5 (including HMAC variant); instead, you should use AES and SHA-256. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The component technologies implemented in IKEv2 are as follows:

- AES-CBC—Advanced Encryption Standard-Cipher Block Chaining
- SHA (HMAC variant)—Secure Hash Algorithm
- Diffie-Hellman—A public-key cryptography protocol
- DES—Data Encryption Standard (No longer recommended)
- MD5 (HMAC [Hash-based Message Authentication Code] variant)—Message digest algorithm 5 (No longer recommended)

For more information about supported standards and component technologies, see the "Supported Standards for Use with IKE" section in the "Configuring Internet Key Exchange for IPsec VPNs" module in the *Internet Key Exchange for IPsec VPNs Configuration Guide*.

Benefits of IKEv2

Dead Peer Detection and Network Address Translation-Traversal

Internet Key Exchange Version 2 (IKEv2) provides built-in support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).

Certificate URLs

Certificates can be referenced through a URL and hash, instead of being sent within IKEv2 packets, to avoid fragmentation.

Denial of Service Attack Resilience

IKEv2 does not process a request until it determines the requester, which addresses to some extent the Denial of Service (DoS) problems in IKEv1, which can be spoofed into performing substantial cryptographic (expensive) processing from false locations.

EAP Support

IKEv2 allows the use of Extensible Authentication Protocol (EAP) for authentication.

Multiple Crypto Engines

If your network has both IPv4 and IPv6 traffic and you have multiple crypto engines, choose one of the following configuration options:

- One engine handles IPv4 traffic and the other engine handles IPv6 traffic.
- One engine handles both IPv4 and IPv6 traffic.

Reliability and State Management (Windowing)

IKEv2 uses sequence numbers and acknowledgments to provide reliability, and mandates some error-processing logistics and shared state management.

Internet Key Exchange Version 2 CLI Constructs

IKEv2 Proposal

An Internet Key Exchange Version 2 (IKEv2) proposal is a collection of transforms used in the negotiation of Internet Key Exchange (IKE) security associations (SAs) as part of the IKE_SA_INIT exchange. The transform types used in the negotiation are as follows:

- Encryption algorithm
- Integrity algorithm
- Pseudo-Random Function (PRF) algorithm
- Diffie-Hellman (DH) group

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 proposal. See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to override the default IKEv2 proposal and to define new proposals.

IKEv2 Policy

An IKEv2 policy contains proposals that are used to negotiate the encryption, integrity, PRF algorithms, and DH group in the IKE_SA_INIT exchange. It can have match statements, which are used as selection criteria to select a policy during negotiation.

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 policy. See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to override the default IKEv2 policy and to define new policies.

IKEv2 Profile

An IKEv2 profile is a repository of nonnegotiable parameters of the IKE SA, such as local or remote identities and authentication methods and services that are available to authenticated peers that match the profile. An IKEv2 profile must be attached to either a crypto map or an IPsec profile on the initiator. An IKEv2 profile is not mandatory on the responder.

IKEv2 Key Ring

An IKEv2 key ring is a repository of symmetric and asymmetric preshared keys and is independent of the IKEv1 key ring. The IKEv2 key ring is associated with an IKEv2 profile and hence supports a set of peers that match the IKEv2 profile. The IKEv2 key ring gets its VPN routing and forwarding (VRF) context from the associated IKEv2 profile.

IKEv2 Smart Defaults

The IKEv2 Smart Defaults feature minimizes the FlexVPN configuration by covering most of the use cases. IKEv2 smart defaults can be customized for specific use cases, though this is not recommended.

See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to modify the default IKEv2 constructs.

The following rules apply to the IKEv2 Smart Defaults feature:

1. A default configuration is displayed in the corresponding **show** command with **default** as a keyword and with no argument. For example, the **show crypto ikev2 proposal default** command displays the default IKEv2 proposal and the **show crypto ikev2 proposal** command displays the default IKEv2 proposal, along with any user-configured proposals.
2. A default configuration is displayed in the **show running-config all** command; it is not displayed in the **show running-config** command.
3. You can modify the default configuration, which is displayed in the **show running-config all** command.
4. A default configuration can be disabled using the **no** form of the command; for example, **no crypto ikev2 proposal default**. A disabled default configuration is not used in negotiation but the configuration is displayed in the **show running-config** command. A disabled default configuration loses any user modification and restores system-configured values.
5. A default configuration can be reenabled using the default form of the command, which restores system-configured values; for example, **default crypto ikev2 proposal**.
6. The default mode for the default transform set is transport; the default mode for all other transform sets is tunnel.

Note Cisco no longer recommends using MD5 (including HMAC variant) and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use SHA-256 and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The following table lists the commands that are enabled with the IKEv2 Smart Defaults feature, along with the default values.

Table 270: IKEv2 Command Defaults

Command Name	Default Values

crypto ikev2 authorization policy	`Device# show crypto ikev2 authorization policy default`
	
	`IKEv2 Authorization policy: default`
	`route set interface`
	`route accept any tag: 1 distance: 2`

crypto ikev2 proposal	`Device# show crypto ikev2 proposal`
	
	`IKEv2 proposal: default`
	`Encryption: AES-CBC-256`
	`Integrity: SHA512 SHA384`
	`PRF: SHA512 SHA384`
	`DH Group: DH_GROUP_256_ECP/Group 19 DH_GROUP_2048_MODP/Group 14 DH_GROUP_521_ECP/Group 21 DH_GROUP_1536_MODP/Group 5`

crypto ikev2 policy	`Device# show crypto ikev2 policy default`
	
	`IKEv2 policy: default`
	`Match fvrf: any`
	`Match address local: any`
	`Proposal: default`

crypto ipsec profile	`Device# show crypto ipsec profile default`
	
	`IPSEC profile default`
	`Security association lifetime: 4608000 kilobytes/3600 seconds`
	`Responder-Only (Y/N): N`
	`PFS (Y/N): N`
	`Transform sets={`
	`default: { esp-aes esp-sha-hmac },`
	`}`

crypto ipsec transform-set	`Device# show crypto ipsec transform-set default`
	
	`Transform set default: { esp-aes esp-sha-hmac }`
	`will negotiate = { Tunnel, },`

Note Before you can use the default IPsec profile, explicitly specify the **crypto ipsec profile** command on a tunnel interface using the **tunnel protection ipsec profile default** command.

Note The 'default' keyword which needs explicit mapping to other CLIs is not supported on a device running on YANG configuration

IKEv2 Suite-B Support

Suite-B is a set of cryptographic algorithms promulgated by the National Security Agency as part of its Cryptographic Modernization Program. Suite-B for Internet Key Exchange (IKE) and IPsec is defined in RFC 4869. The Suite-B components are as follows:

- Advanced Encryption Standard (AES) 128- and 256-bit keys configured in the IKEv2 proposal. For data traffic, AES should be used in Galois Counter Mode (GCM) that is configured in the IPsec transform set.
- Elliptic Curve Digital Signature Algorithm (ECDSA) configured in the IKEv2 profile.
- Secure Hashing Algorithm 2 (SHA-256 and SHA-384) configured in the IKEv2 proposal and IPsec transform set.

Suite-B requirements comprise four user-interface suites of cryptographic algorithms for use with IKE and IPsec. Each suite consists of an encryption algorithm, a digital-signature algorithm, a key-agreement algorithm, and a hash- or message-digest algorithm. See the "Configuring Security for VPNs with IPsec" feature module for detailed information about Cisco Suite-B support.

AES-GCM Support

An authenticated encryption algorithm provides a combined functionality of encryption and integrity. Such algorithms are called combined mode algorithms. The Support of AES-GCM as an IKEv2 Cipher on IOS feature provides the use of authenticated encryption algorithms for encrypted messages in IKEv2 protocol by adding the Advanced Encryption Standard in Galois/Counter Mode (AES-GCM). AES-GCM supports the key size of 128- and 256-bits—AES-GCM-128 and AES-GCM-256.

Note If AES-GCM is the only encryption algorithm, integrity algorithms cannot be added to the proposal.

Auto Tunnel Mode Support in IKEv2

When configuring a VPN headend in a multiple vendor scenario, you must be aware of the technical details of the peer or responder. For example, some devices may use IPsec tunnels while others may use generic routing encapsulation (GRE) or IPsec tunnel, and sometimes, a tunnel may be IPv4 or IPv6. In the last case, you must configure an Internet Key Exchange (IKE) profile and a virtual template.

The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. This feature is useful on dual stack hubs aggregating multivendor remote access, such as Cisco AnyConnect VPN Client, Microsoft Windows7 Client, and so on.

Note The Tunnel Mode Auto Selection feature eases the configuration for a responder only. The tunnel must be statically configured for an initiator.

The Tunnel Mode Auto Selection feature can be activated using the **auto mode** keywords in the **virtual-template** command in the IKEv2 profile configuration.

How to Configure Internet Key Exchange Version 2

Configuring Basic Internet Key Exchange Version 2 CLI Constructs

To enable IKEv2 on a crypto interface, attach an Internet Key Exchange Version 2 (IKEv2) profile to the crypto map or IPsec profile applied to the interface. This step is optional on the IKEv2 responder.

Note The difference between IKEv1 and IKEv2 is that you need not enable IKEv1 on individual interfaces because IKEv1 is enabled globally on all interfaces on a device.

Perform the following tasks to manually configure basic IKEv2 constructs:

Configuring the IKEv2 Keyring

Perform this task to configure the IKEv2 key ring if the local or remote authentication method is a preshared key.

IKEv2 key ring keys must be configured in the peer configuration submode that defines a peer subblock. An IKEv2 key ring can have multiple peer subblocks. A peer subblock contains a single symmetric or asymmetric key pair for a peer or peer group identified by any combination of the hostname, identity, and IP address.

IKEv2 key rings are independent of IKEv1 key rings. The key differences are as follows:

- IKEv2 key rings support symmetric and asymmetric preshared keys.
- IKEv2 key rings do not support Rivest, Shamir, and Adleman (RSA) public keys.
- IKEv2 key rings are specified in the IKEv2 profile and are not looked up, unlike IKEv1, where keys are looked up on receipt of MM1 to negotiate the preshared key authentication method. The authentication method is not negotiated in IKEv2.
- IKEv2 key rings are not associated with VPN routing and forwarding (VRF) during configuration. The VRF of an IKEv2 key ring is the VRF of the IKEv2 profile that refers to the key ring.
- A single key ring can be specified in an IKEv2 profile, unlike an IKEv1 profile, which can specify multiple key rings.
- A single key ring can be specified in more than one IKEv2 profile, if the same keys are shared across peers matching different profiles.
- An IKEv2 key ring is structured as one or more peer subblocks.

On an IKEv2 initiator, the IKEv2 key ring key lookup is performed using the peer's hostname or the address, in that order. On an IKEv2 responder, the key lookup is performed using the peer's IKEv2 identity or the address, in that order.

Note You cannot configure the same identity in more than one peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 keyring** *keyring-name*
4. **peer** *name*
5. **description** *line-of-description*
6. **hostname** *name*
7. **address** {*ipv4-address* [*mask*] | *ipv6-address prefix*}
8. **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
9. **pre-shared-key** {**local** | **remote**} [**0** | **6**] *line* **hex** *hexadecimal-string*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 keyring** *keyring-name* **Example:** `Device(config)# crypto ikev2 keyring kyr1`	Defines an IKEv2 key ring and enters IKEv2 key ring configuration mode.
Step 4	**peer** *name* **Example:** `Device(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group and enters IKEv2 key ring peer configuration mode.
Step 5	**description** *line-of-description* **Example:** `Device(config-ikev2-keyring-peer)# description this is the first peer`	(Optional) Describes the peer or peer group.
Step 6	**hostname** *name* **Example:** `Device(config-ikev2-keyring-peer)# hostname host1`	Specifies the peer using a hostname.

	Command or Action	Purpose
Step 7	**address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} **Example:** `Device(config-ikev2-keyring-peer)# address 10.0.0.1 255.255.255.0`	Specifies an IPv4 or IPv6 address or range for the peer. **Note** This IP address is the IKE endpoint address and is independent of the identity address.

Step 8	**identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
	Example:
	`Device(config-ikev2-keyring-peer)# identity address 10.0.0.5`
		Identifies the IKEv2 peer through the following identities:
		• E-mail
		• Fully qualified domain name (FQDN) .
		Note When FQDN is used to identify the peer in the keyring configuration, use the IP address of the peer along with the FQDN
		`crypto ikev2 keyring key1`
		`peer headend-1`
		`address 1.1.1.1 >>>>>>>>>`
		`identity fqdn NFVIS-headend-1.cisco.com`
		`pre-shared-key Cisco123`
		• IPv4 or IPv6 address
		• Key ID
		Note The identity is available for key lookup on the IKEv2 responder only.

Step 9	**pre-shared-key** {**local** | **remote**} [**0** | **6**] *line* **hex** *hexadecimal-string* **Example:** `Device(config-ikev2-keyring-peer)# pre-shared-key local key1`	Specifies the preshared key for the peer.
Step 10	**end** **Example:** `Device(config-ikev2-keyring-peer)# end`	Exits IKEv2 key ring peer configuration mode and returns to privileged EXEC mode.

What to Do Next

After configuring the IKEv2 key ring, configure the IKEv2 profile. For more information, see the "Configuring IKEv2 Profile (Basic)" section.

Configuring an IKEv2 Profile (Basic)

Perform this task to configure the mandatory commands for an IKEv2 profile.

An IKEv2 profile is a repository of nonnegotiable parameters of the IKE security association (SA) (such as local or remote identities and authentication methods) and services available to authenticated peers that match the profile. An IKEv2 profile must be configured and associated with either a crypto map or an IPsec profile

on the IKEv2 initiator. Use the **set ikev2-profile** *profile-name* command to associate a profile with a crypto map or an IPsec profile. To disassociate the profile, use the **no** form of the command.

The following rules apply to match statements:

- An IKEv2 profile must contain a match identity or a match certificate statement; otherwise, the profile is considered incomplete and is not used. An IKEv2 profile can have more than one match identity or match certificate statements.
- An IKEv2 profile must have a single match Front Door VPN routing and forwarding (FVRF) statement.
- When a profile is selected, multiple match statements of the same type are logically ORed, and multiple match statements of different types are logically ANDed.
- The match identity and match certificate statements are considered to be the same type of statements and are ORed.
- Configuration of overlapping profiles is considered a misconfiguration. In the case of multiple profile matches, no profile is selected.

Use the **show crypto ikev2 profile** *profile-name* command to display the IKEv2 profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **description** *line-of-description*
5. **aaa accounting** {**psk** | **cert** | **eap**} *list-name*
6. **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}
7. **dpd** *interval retry-interval* {**on-demand** | **periodic**}
8. **dynamic**
9. **identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*}
10. **initial-contact force**
11. **ivrf** *name*
12. **keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] }
13. **lifetime** *seconds*
14. **match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*}
15. **nat keepalive** *seconds*
16. **pki trustpoint** *trustpoint-label* [**sign** | **verify**]
17. **virtual-template** *number* **mode auto**
18. **shutdown**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters the IKEv2 profile configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-ikev2-profile)# description This is an IKEv2 profile`	(Optional) Describes the profile.
Step 5	**aaa accounting** {**psk** | **cert** | **eap**} *list-name* **Example:** `Device(config-ikev2-profile)# aaa accounting eap list1`	(Optional) Enables authentication, authorization, and accounting (AAA) accounting method lists for IPsec sessions. **Note** If the **psk**, **cert**, or **eap** keyword is not specified, the AAA accounting method list is used irrespective of the peer authentication method.
Step 6	**authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}} **Example:** `Device(config-ikev2-profile)# authentication local ecdsa-sig`	Specifies the local or remote authentication method. • **rsa-sig**—Specifies RSA-sig as the authentication method. • **pre-share**—Specifies the preshared key as the authentication method. • **ecdsa-sig**—Specifies ECDSA-sig as the authentication method. • **eap**—Specifies EAP as the remote authentication method. • **query-identity**—Queries the EAP identity from the peer. • **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response.

	Command or Action	Purpose
		Note You can specify only one local authentication method but multiple remote authentication methods.
Step 7	**dpd** *interval retry-interval* {**on-demand** | **periodic**} **Example:** `Device(config-ikev2-profile)# dpd 30 6 on-demand`	This step is optional. Configures Dead Peer Detection (DPD) globally for peers matching the profile. By default, the Dead Peer Detection (DPD) is disabled. **Note** In the example in thisstep, the first DPD issent after 30 seconds when there is no incoming ESP traffic. After waiting for 6 seconds (which is the specified retry interval), DPD retries are sent agressively 5 timesin intervals of 6 seconds each. So, a total of 66 seconds (30 + 6 + 6 * 5 = 66) elapses before a crypto session is torn down because of DPD.
Step 8	**dynamic** **Example:** `Device(config-ikev2-profile)# dynamic`	Configures a dynamic IKEv2 profile. This keyword has been introduced in the Cisco IOS XE 17.2.1 release. **Note** When you configure a dynamic profile, you cannot configure local or remote authentication and identity using the command line interface.
Step 9	**identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# identity local email abc@example.com`	This is an optional step. Specifies the local IKEv2 identity type. **Note** If the local authentication method is a preshared key, the default local identity is the IP address. If the local authentication method is a Rivest, Shamir, and Adleman (RSA) signature, the default local identity is a Distinguished Name.
Step 10	**initial-contact force** **Example:** `Device(config-ikev2-profile)# initial-contact force`	Enforces initial contact processing if the initial contact notification is not received in the IKE_AUTH exchange.
Step 11	**ivrf** *name* **Example:** `Device(config-ikev2-profile)# ivrf vrf1`	This is an optional step. Specifies a user-defined VPN routing and forwarding (VRF) or global VRF if the IKEv2 profile is attached to a crypto map. • If you use the IKEv2 profile for tunnel protection, you must configure the Inside VRF (IVRF) for the tunnel interface on the tunnel interface. **Note** IVRF specifies the VRF for cleartext packets. The default value for IVRF is FVRF.

	Command or Action	Purpose
Step 12	**keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] } **Example:** `Device(config-ikev2-profile)# keyring aaa keyring1 name-mangler mangler1`	Specifies the local or AAA-based key ring that must be used with the local and remote preshared key authentication method. **Note** You can specify only one key ring. Local AAA is not supported for AAA-based preshared keys. **Note** Depending on your release, the **local** keyword and the **name-mangler** *mangler-name* keyword-argument pair should be used. **Note** When using AAA, the default password for a Radius access request is "cisco". You can use the **password** keyword within the **keyring** command to change the password. **Note** To remove the keyring from the IKEv2 profile, use the **no keyring** {**aaa** | **local** | **ppk**} *keyring-name* command.
Step 13	**lifetime** *seconds* **Example:** `Device(config-ikev2-profile)# lifetime 1000`	Specifies the lifetime, in seconds, for the IKEv2 SA.
Step 14	**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# match address local interface Ethernet 2/0`	Uses match statements to select an IKEv2 profile for a peer.
Step 15	**nat keepalive** *seconds* **Example:** `Device(config-ikev2-profile)# nat keepalive 500`	(Optional) Enables NAT keepalive and specifies the duration in seconds. • By default, NAT is disabled.
Step 16	**pki trustpoint** *trustpoint-label* [**sign** | **verify**] **Example:** `Device(config-ikev2-profile)# pki trustpoint tsp1 sign`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method. **Note** If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification.

	Command or Action	Purpose
		Note In contrast to IKEv1, a trustpoint must be configured in an IKEv2 profile for certificate-based authentication to succeed. There is no fallback for globally configured trustpoints if this command is not present in the configuration. The trustpoint configuration applies to the IKEv2 initiator and responder.
Step 17	**virtual-template** *number* **mode auto** **Example:** `Device(config-ikev2-profile)# virtual-template 1 mode auto`	This is an optional step. Specifies the virtual template for cloning a virtual access interface (VAI). • **mode auto** - Enables the tunnel mode auto selection feature. **Note** For the IPsec Dynamic Virtual Tunnel Interface (DVTI), a virtual template must be specified in an IKEv2 profile, without which an IKEv2 session is not initiated.
Step 18	**shutdown** **Example:** `Device(config-ikev2-profile)# shutdown`	(Optional) Shuts down the IKEv2 profile.
Step 19	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits the IKEv2 profile configuration mode and returns to the privileged EXEC mode.

Configuring Advanced Internet Key Exchange Version 2 CLI Constructs

This section describes the global IKEv2 CLI constructs and how to override the IKEv2 default CLI constructs. IKEv2 smart defaults support most use cases and hence, we recommend that you override the defaults only if they are required for specific use cases not covered by the defaults.

Perform the following tasks to configure advanced IKEv2 CLI constructs:

Configuring Global IKEv2 Options

Perform this task to configure global IKEv2 options that are independent of peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 certificate-cache** *number-of-certificates*
4. **crypto ikev2 cookie-challenge** *number*
5. **crypto ikev2 diagnose error** *number*
6. **crypto ikev2 dpd** *interval retry-interval* {**on-demand** | **periodic**}

7. **crypto ikev2 http-url cert**
8. **crypto ikev2 limit** {**max-in-negotiation-sa** *limit* | **max-sa** *limit*}
9. **crypto ikev2 nat keepalive** *interval*
10. **crypto ikev2 window** *size*
11. **crypto logging ikev2**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 certificate-cache** *number-of-certificates* **Example:** `Device(config)# crypto ikev2 certificate-cache 750`	Defines the cache size for storing certificates fetched from HTTP URLs.
Step 4	**crypto ikev2 cookie-challenge** *number* **Example:** `Device(config)# crypto ikev2 cookie-challenge 450`	Enables an IKEv2 cookie challenge only when the number of half-open security associations (SAs) exceeds the configured number. • Cookie challenge is disabled by default.
Step 5	**crypto ikev2 diagnose error** *number* **Example:** `Device(config)# crypto ikev2 diagnose error 500`	Enables IKEv2 error diagnostics and defines the number of entries in the exit path database. • IKEv2 error diagnostics is disabled by default.
Step 6	**crypto ikev2 dpd** *interval retry-interval* {**on-demand** | **periodic**} **Example:** `Device(config)# crypto ikev2 dpd 30 6 on-demand`	Allows live checks for peers as follows: • Dead Peer Detection (DPD) is disabled by default. **Note** In the example in this step, the first DPD is sent after 30 seconds when there is no incoming ESP traffic. After waiting for 6 seconds (which is the specified retry interval), DPD retries are sent agressively 5 times in intervals of 6 seconds each. So, a total of 66 seconds (30 + 6 + 6 * 5 = 66) elapses before a crypto session is torn down because of DPD.
Step 7	**crypto ikev2 http-url cert**	Enables the HTTP CERT support.

	Command or Action	Purpose
	Example: `Device(config)# crypto ikev2 http-url cert`	• HTTP CERT is disabled by default.
Step 8	**crypto ikev2 limit** {**max-in-negotiation-sa** *limit* | **max-sa** *limit*} **Example:**	Enables connection admission control (CAC). • Connection admission control is enabled by default.
Step 9	**crypto ikev2 nat keepalive** *interval* **Example:** `Device(config)# crypto ikev2 nat keepalive 500`	Enables the Network Address Translation (NAT) keepalive that prevents the deletion of NAT entries in the absence of any traffic when there is NAT between Internet Key Exchange (IKE) peers. • NAT keepalive is disabled by default.
Step 10	**crypto ikev2 window** *size* **Example:** `Device(config)# crypto ikev2 window 15`	Allows multiple IKEv2 request-response pairs in transit. • The default window size is 5.
Step 11	**crypto logging ikev2** **Example:** `Device(config)# crypto logging ikev2`	Enables IKEv2 syslog messages. • IKEv2 syslog messages are disabled by default.
Step 12	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IKEv2 Fragmentation

Perform this task to enable automatic fragmentation of large IKEv2 packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 fragmentation** [**mtu** *mtu-size*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**crypto ikev2 fragmentation** [**mtu** *mtu-size*] **Example:** `Device(config)# crypto ikev2 fragmentation mtu 100`	Configures IKEv2 fragmentation. • The MTU range is from 96 to 1500 bytes. The default MTU size is 576 for IPv4 packets and 1280 bytes for IPv6 packets. **Note** The MTU size refers to the IP or UDP encapsulated IKEv2 packets.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IKEv2 Proposal

Refer to the "IKEv2 Smart Defaults" section for information on the default IKEv2 proposal.

Perform this task to override the default IKEv2 proposal or to manually configure the proposals if you do not want to use the default proposal.

An IKEv2 proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, the default proposal in the default IKEv2 policy is used in negotiation.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Although the IKEv2 proposal is similar to the **crypto isakmp policy** command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuring one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 proposal** *name*
4. **encryption** *encryption-type...*
5. **integrity** *integrity-type...*
6. **group** *group-type...*
7. **prf** *prf-algorithm*
8. **end**

9. **show crypto ikev2 proposal** [*name* | **default**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 proposal** *name* **Example:** `Device(config)# crypto ikev2 proposal proposal1`	Overrides the default IKEv2 proposal, defines an IKEv2 proposal name, and enters IKEv2 proposal configuration mode.
Step 4	**encryption** *encryption-type...* **Example:** `Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-192`	Specifies one or more transforms of the encryption type, which are as follows: • **3des** (No longer recommended) • **aes-cbc-128** • **aes-cbc-192** • **aes-cbc-256** • **aes-gcm-128** • **aes-gcm-256**
Step 5	**integrity** *integrity-type...* **Example:** `Device(config-ikev2-proposal)# integrity sha1`	Specifies one or more transforms of the integrity algorithm type, which are as follows: • The **md5** keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended) • The **sha1** keyword specifies SHA-1 (HMAC variant) as the hash algorithm. • The **sha256** keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm. • The **sha384** keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm. • The **sha512** keyword specifies SHA-2 family 512-bit (HMAC variant) as the hash algorithm. **Note** An integrity algorithm type cannot be specified if you specify Advanced Encryption Standard (AES) in Galois/Counter Mode (AES GCM) as the encryption type.

	Command or Action	Purpose
Step 6	**group** *group-type...* **Example:** `Device(config-ikev2-proposal)# group 14`	Specifies the Diffie-Hellman (DH) group identifier. • The default DH group identifiers are group 2 and 5 in the IKEv2 proposal. • **1**—768-bit DH (No longer recommended). • **2**—1024-bit DH (No longer recommended). • **5**—1536-bit DH (No longer recommended). • **14**—Specifies the 2048-bit DH group. • **15**—Specifies the 3072-bit DH group. • **16**—Specifies the 4096-bit DH group. • **19**—Specifies the 256-bit elliptic curve DH (ECDH) group. • **20**—Specifies the 384-bit ECDH group. • **24**—Specifies the 2048-bit DH group. The group chosen must be strong enough (have enough bits) to protect the IPsec keys during negotiation. A generally accepted guideline recommends the use of a 2048-bit group after 2013 (until 2030). Either group 14 or group 24 can be selected to meet this guideline. Even if a longer-lived security method is needed, the use of Elliptic Curve Cryptography is recommended, but group 15 and group 16 can also be considered.
Step 7	**prf** *prf-algorithm* **Example:** `Device(config-ikev2-proposal)# prf sha256 sha512`	Specifies one or more of the Pseudo-Random Function (PRF) algorithm, which are as follows: • **md5** • **sha1** • **sha256** • **sha384** • **sha512** **Note** This step is mandatory if the encryption type is AES-GCM—**aes-gmc-128** or **aes-gmc-256**. If the encryption algorithm is not AES-GCM, the PRF algorithm is the same as the specified integrity algorithm. However, you can specify a PRF algorithm, if required.
Step 8	**end** **Example:** `Device(config-ikev2-proposal)# end`	Exits IKEv2 proposal configuration mode and returns to privileged EXEC mode.
Step 9	**show crypto ikev2 proposal** [*name* | **default**] **Example:** `Device# show crypto ikev2 proposal default`	(Optional) Displays the IKEv2 proposal.

What to Do Next

After you create the IKEv2 proposal, attach it to a policy so that the proposal is picked for negotiation. For information about completing this task, see the "Configuring IKEv2 Policy" section.

Configuring IKEv2 Policies

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 policy.

Perform this task to override the default IKEv2 policy or to manually configure the policies if you do not want to use the default policy.

An IKEv2 policy must contain at least one proposal to be considered as complete and can have match statements, which are used as selection criteria to select a policy for negotiation. During the initial exchange, the local address (IPv4 or IPv6) and the Front Door VRF (FVRF) of the negotiating SA are matched with the policy and the proposal is selected.

The following rules apply to the match statements:

- An IKEv2 policy without any match statements will match all peers in the global FVRF.
- An IKEv2 policy can have only one match FVRF statement.
- An IKEv2 policy can have one or more match address local statements.
- When a policy is selected, multiple match statements of the same type are logically ORed and match statements of different types are logically ANDed.
- There is no precedence between match statements of different types.
- Configuration of overlapping policies is considered a misconfiguration. In the case of multiple, possible policy matches, the first policy is selected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 policy** *name*
4. **proposal** *name*
5. **match fvrf** {*fvrf-name* | **any**}
6. **match address local** {*ipv4-address* | *ipv6-address*}
7. **end**
8. **show crypto ikev2 policy** [*policy-name* | **default**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**crypto ikev2 policy** *name* **Example:** `Device(config)# crypto ikev2 policy policy1`	Overrides the default IKEv2 policy, defines an IKEv2 policy name, and enters IKEv2 policy configuration mode.
Step 4	**proposal** *name* **Example:** `Device(config-ikev2-policy)# proposal proposal1`	Specifies the proposals that must be used with the policy. • The proposals are prioritized in the order of listing. **Note** You must specify at least one proposal. You can specify additional proposals with each proposal in a separate statement.
Step 5	**match fvrf** {*fvrf-name* | **any**} **Example:** `Device(config-ikev2-policy)# match fvrf any`	(Optional) Matches the policy based on a user-configured FVRF or any FVRF. • The default is global FVRF. **Note** The **match fvrf any** command must be explicitly configured in order to match any VRF. The FVRF specifies the VRF in which the IKEv2 packets are negotiated.
Step 6	**match address local** {*ipv4-address* | *ipv6-address*} **Example:** `Device(config-ikev2-policy)# match address local 10.0.0.1`	(Optional) Matches the policy based on the local IPv4 or IPv6 address. • The default matches all the addresses in the configured FVRF.
Step 7	**end** **Example:** `Device(config-ikev2-policy)# end`	Exits IKEv2 policy configuration mode and returns to privileged EXEC mode.
Step 8	**show crypto ikev2 policy** [*policy-name* | **default**] **Example:** `Device# show crypto ikev2 policy policy1`	(Optional) Displays the IKEv2 policy.

Configuration Examples for Internet Key Exchange Version 2

Configuration Examples for Basic Internet Key Exchange Version 2 CLI Constructs

Example: Configuring the IKEv2 Key Ring

Example: IKEv2 Key Ring with Multiple Peer Subblocks

The following example shows how to configure an Internet Key Exchange Version 2 (IKEv2) key ring with multiple peer subblocks:

```
crypto ikev2 keyring keyring-1
 peer peer1
  description peer1
  address 209.165.200.225 255.255.255.224
  pre-shared-key key-1
 peer peer2
  description peer2
  hostname peer1.example.com
  pre-shared-key key-2
 peer peer3
  description peer3
  hostname peer3.example.com
  identity key-id abc
  address 209.165.200.228 255.255.255.224
  pre-shared-key key-3
```

Example: IKEv2 Key Ring with Symmetric Preshared Keys Based on an IP Address

The following example shows how to configure an IKEv2 key ring with symmetric preshared keys based on an IP address. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer1
  description peer1
  address 209.165.200.225 255.255.255.224
  pre-shared-key key1
```

The following is the responder's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer2
  description peer2
  address 209.165.200.228 255.255.255.224
  pre-shared-key key1
```

Example: IKEv2 Key Ring with Asymmetric Preshared Keys Based on an IP Address

The following example shows how to configure an IKEv2 key ring with asymmetric preshared keys based on an IP address. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
```

```
 peer peer1
  description peer1 with asymmetric keys
  address 209.165.200.225 255.255.255.224
  pre-shared-key local key1
  pre-shared-key remote key2
```

The following is the responder's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer2
  description peer2 with asymmetric keys
  address 209.165.200.228 255.255.255.224
  pre-shared-key local key2
  pre-shared-key remote key1
```

Example: IKEv2 Key Ring with Asymmetric Preshared Keys Based on a Hostname

The following example shows how to configure an IKEv2 key ring with asymmetric preshared keys based on the hostname. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
 peer host1
  description host1 in example domain
  hostname host1.example.com
  pre-shared-key local key1
  pre-shared-key remote key2
```

The following is the responder's keyring:

```
crypto ikev2 keyring keyring-1
 peer host2
  description host2 in abc domain
  hostname host2.example.com
  pre-shared-key local key2
  pre-shared-key remote key1
```

Example: IKEv2 Key Ring with Symmetric Preshared Keys Based on an Identity

The following example shows how to configure an IKEv2 key ring with symmetric preshared keys based on an identity:

```
crypto ikev2 keyring keyring-4
 peer abc
  description example domain
  identity fqdn example.com
  pre-shared-key abc-key-1
 peer user1
  description user1 in example domain
  identity email user1@example.com
  pre-shared-key abc-key-2
 peer user1-remote
  description user1 example remote users
  identity key-id example
  pre-shared-key example-key-3
```

Example: IKEv2 Key Ring with a Wildcard Key

The following example shows how to configure an IKEv2 key ring with a wildcard key:

```
crypto ikev2 keyring keyring-1
 peer cisco
  description example domain
  address 0.0.0.0 0.0.0.0
  pre-shared-key example-key
```

Example: Matching a Key Ring

The following example shows how a key ring is matched:

```
crypto ikev2 keyring keyring-1
 peer cisco
  description example.com
  address 0.0.0.0 0.0.0.0
  pre-shared-key xyz-key
 peer peer1
  description abc.example.com
  address 10.0.0.0 255.255.0.0
  pre-shared-key abc-key
 peer host1
  description host1@abc.example.com
  address 10.0.0.1
  pre-shared-key host1-example-key
```

In the example shown, the key lookup for peer 10.0.0.1 first matches the wildcard key example-key, then the prefix key example-key, and finally the host key host1-example-key. The best match host1-example-key is used.

```
crypto ikev2 keyring keyring-2
 peer host1
 description host1 in abc.example.com sub-domain
 address 10.0.0.1
 pre-shared-key host1-example-key
 peer host2
  description example domain
  address 0.0.0.0 0.0.0.0
  pre-shared-key example-key
```

In the example shown, the key lookup for peer 10.0.0.1 would first match the host key host1-abc-key. Because this is a specific match, no further lookup is performed.

Example: Configuring the Profile

Example: IKEv2 Profile Matched on Remote Identity

The following profile supports peers that identify themselves using fully qualified domain name (FQDN) example.com and authenticate with the RSA signature using trustpoint-remote. The local node authenticates itself with a preshared key using keyring-1.

```
crypto ikev2 profile profile2
 match identity remote fqdn example.com
 identity local email router2@example.com
 authentication local pre-share
 authentication remote rsa-sig
 keyring keyring-1
 pki trustpoint trustpoint-remote verify
 lifetime 300
 dpd 10 5 on-demand
 virtual-template 1
```

Example: IKEv2 Profile Supporting Two Peers

The following example shows how to configure an IKEv2 profile supporting two peers that use different authentication methods:

```
crypto ikev2 profile profile2
 match identity remote email user1@example.com
 match identity remote email user2@example.com
 identity local email router2@cisco.com
 authentication local rsa-sig
 authentication remote pre-share
 authentication remote rsa-sig
 keyring keyring-1
 pki trustpoint trustpoint-local sign
 pki trustpoint trustpoint-remote verify
 lifetime 300
 dpd 10 5 on-demand
 virtual-template 1
```

Example: Configuring FlexVPN with Dynamic Routing Using Certificates and IKEv2 Smart Defaults

The following examples show a connection between a branch device (initiator, using a static virtual tunnel interface [sVTI]) and a central device (responder, using a dynamic virtual tunnel interface [dVTI]) with dynamic routing over the tunnel. The example uses IKEv2 smart defaults, and the authentication is performed using certificates (RSA signatures).

Note A RSA modulus size of 2048 is recommended.

The peers use the FQDN as their IKEv2 identity, and the IKEv2 profile on the responder matches the domain in the identity FQDN.

The configuration on the initiator (branch device) is as follows:

```
hostname branch
ip domain name cisco.com
!
crypto ikev2 profile branch-to-central
 match identity remote fqdn central.cisco.com
 identity local fqdn branch.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
!
crypto ipsec profile svti
 set ikev2-profile branch-to-central
!
interface Tunnel0
 ip address 172.16.0.101 255.255.255.0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 10.0.0.100
 tunnel protection ipsec profile svti
!
interface Ethernet0/0
 ip address 10.0.0.101 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.101.1 255.255.255.0
```

```
!
router rip
 version 2
 passive-interface Ethernet1/0
 network 172.16.0.0
 network 192.168.101.0
 no auto-summary
```

The configuration on the responder (central router) is as follows:

```
hostname central
ip domain name cisco.com
!
crypto ikev2 profile central-to-branch
 match identity remote fqdn domain cisco.com
 identity local fqdn central.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 virtual-template 1
!
interface Loopback0
 ip address 172.16.0.100 255.255.255.0
!
interface Ethernet0/0
 ip address 10.0.0.100 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.100.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile default
!
router rip
 version 2
 passive-interface Ethernet1/0
 network 172.16.0.0
 network 192.168.100.0
 no auto-summary
```

Configuration Examples for Advanced Internet Key Exchange Version 2 CLI Constructs

Example: Configuring the Proposal

Example: IKEv2 Proposal with One Transform for Each Transform Type

This example shows how to configure an IKEv2 proposal with one transform for each transform type:

```
crypto ikev2 proposal proposal-1
 encryption aes-cbc-128
 integrity sha1
 group 14
```

Example: IKEv2 Proposal with Multiple Transforms for Each Transform Type

This example shows how to configure an IKEv2 proposal with multiple transforms for each transform type:

```
crypto ikev2 proposal proposal-2
 encryption aes-cbc-128 aes-cbc-192
 integrity sha1
 group 14
```

Note Cisco no longer recommends using 3DES, MD5 (including HMAC variant), and Diffie-Hellman(DH) groups 1, 2 and 5; instead, you should use AES, SHA-256 and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The IKEv2 proposal proposal-2 shown translates to the following prioritized list of transform combinations:

- aes-cbc-128, sha1, 14
- aes-cbc-192, sha1, 14

Example: IKEv2 Proposals on the Initiator and Responder

The following example shows how to configure IKEv2 proposals on the initiator and the responder. The proposal on the initiator is as follows:

```
crypto ikev2 proposal proposal-1
 encryption aes-cbc-192 aes-cbc-128
 integrity sha-256 sha1
 group 14 24
```

The proposal on the responder is as follows:

```
crypto ikev2 proposal proposal-2
 encryption aes-cbc-128 aes-cbc-192
 peer
 integrity sha1 sha-256
 group 24 14
```

The selected proposal will be as follows:

```
encryption aes-cbc-128
integrity sha1
group 14
```

In the proposals shown for the initiator and responder, the initiator and responder have conflicting preferences. In this case, the initiator is preferred over the responder.

Example: Configuring the Policy

Example: IKEv2 Policy Matched on a VRF and Local Address

The following example shows how an IKEv2 policy is matched based on a VRF and local address:

```
crypto ikev2 policy policy2
 match vrf vrf1
```

```
 match local address 10.0.0.1
 proposal proposal-1
```

Example: IKEv2 Policy with Multiple Proposals That Match All Peers in a Global VRF

The following example shows how an IKEv2 policy with multiple proposals matches the peers in a global VRF:

```
crypto ikev2 policy policy2
 proposal proposal-A
 proposal proposal-B
 proposal proposal-B
```

Example: IKEv2 Policy That Matches All Peers in Any VRF

The following example shows how an IKEv2 policy matches the peers in any VRF:

```
crypto ikev2 policy policy2
 match vrf any
 proposal proposal-1
```

Example: Matching a Policy

Do not configure overlapping policies. If there are multiple possible policy matches, the best match is used, as shown in the following example:

```
crypto ikev2 policy policy1
 match fvrf fvrf1
crypto ikev2 policy policy2
 match fvrf fvff1
 match local address 10.0.0.1
```

The proposal with FVRF as fvrf1 and the local peer as 10.0.0.1 matches policy1 and policy2, but policy2 is selected because it is the best match.

Where to Go Next

After configuring IKEv2, proceed to configure IPsec VPNs. For more information, see the “Configuring Security for VPNs with IPsec” module.

Additional References for Configuring Internet Key Exchange Version 2 (IKEv2)

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	Configuring Security for VPNs with IPsec
Suite-B ESP transforms	Configuring Security for VPNs with IPsec
Suite-B SHA-2 family (HMAC variant) and elliptic curve (EC) key pair configuration	Configuring Internet Key Exchange for IPsec VPNs
Suite-B elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	Configuring Internet Key Exchange for IPsec VPNs
Suite-B support for certificate enrollment for a PKI	Configuring Certificate Enrollment for a PKI
Supported standards for use with IKE	Internet Key Exchange for IPsec VPNs Configuration Guide
Recommended cryptographic algorithms	Next Generation Encryption

RFCs

RFC	Title
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4869	*Suite B Cryptographic Suites for IPsec*
RFC 5685	*Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Internet Key Exchange Version 2 (IKEv2)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 271: Feature Information for Configuring Internet Key Exchange Version 2 (IKEv2)

Feature Name	Releases	Feature Information
IPv6 Support for IPsec and IKEv2		This feature allows IPv6 addresses to be added to IPsec and IKEv2 protocols. The following commands were introduced or modified: **address (IKEv2 keyring), identity (IKEv2 keyring), identity local, match (IKEv2 policy), match (IKEv2 profile), show crypto ikev2 session, show crypto ikev2 sa, show crypto ikev2 profile, show crypto ikev2 policy, debug crypto condition, clear crypto ikev2 sa.**
Suite-B Support in IOS SW Crypto		Suite-B adds support for the SHA-2 family (HMAC variant) hash algorithm used to authenticate packet data and verify the integrity verification mechanisms for the IKEv2 proposal configuration. HMAC is a variant that provides an additional level of hashing. Suite-B also allows the Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig), as defined in RFC 4754, to be the authentication method for IKEv2. Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite is consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec module for more information about Cisco IOS Suite-B support. The following commands were introduced or modified: **authentication, group, identity (IKEv2 profile), integrity, match (IKEv2 profile).**
Support of AES-GCM as an IKEv2 Cipher on IOS		The AES-GCM Support on IKEv2 feature describes the use of authenticated encryption algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) protocol by adding the Advanced Encryption Standard (AES) in Galois/Counter Mode (AES-GCM). The following commands were introduced or modified: **encryption (IKEv2 proposal), prf, show crypto ikev2 proposal.**

Feature Name	Releases	Feature Information
Tunnel Mode Auto Selection		The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. The following commands were introduced or modified: **virtual-template (IKEv2 profile), show crypto ikev2 profile.**

CHAPTER 206

Configuring Quantum-Safe Encryption Using Postquantum Preshared Keys

This module describes quantum-safe encryption using Postquantum Preshared Keys (PPK). This feature implements RFC 8784 and Cisco Secure Key Integration Protocol (SKIP) for quantum-safe encryption of IKEv2 and IPsec packets using PPKs.

Restrictions for Quantum-Safe Encryption Using Postquantum Preshared Keys

- The Quantum-Safe Encryption Using Postquantum Preshared Keys feature is applicable to all IKEv2 and IPsec VPNs such as, FlexVPN (SVTI-DVTI) and DMVPN, except for GETVPN.

Supported Platforms

The Quantum-Safe Encryption Using Postquantum Preshared Keys feature is available on the following platforms:

From Cisco IOS XE Release 17.11	From Cisco IOS XE Release 17.12
Cisco Catalyst 8000V Edge Software	Cisco 1000 Series Integrated Services Routers
Cisco Catalyst 8300 Series Edge Platforms	Cisco Catalyst 8500 Series Edge Platforms
Cisco ASR 1000 Series Aggregation Services Routers	

Information About Quantum-Safe Encryption Using Postquantum Preshared Keys

The following sections provide detailed information relating to the Quantum-Safe Encryption Using Postquantum Preshared Keys feature.

Impact of Quantum Computers on Cryptography

Quantum computers pose a serious challenge to the cryptographic algorithms and protocols deployed widely today. A quantum computer can solve Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) problems in polynomial time, and this can compromise the security of existing IKEv2 systems. A man-in-the-middle storing the VPN communications today can decrypt them later when a quantum computer is available.

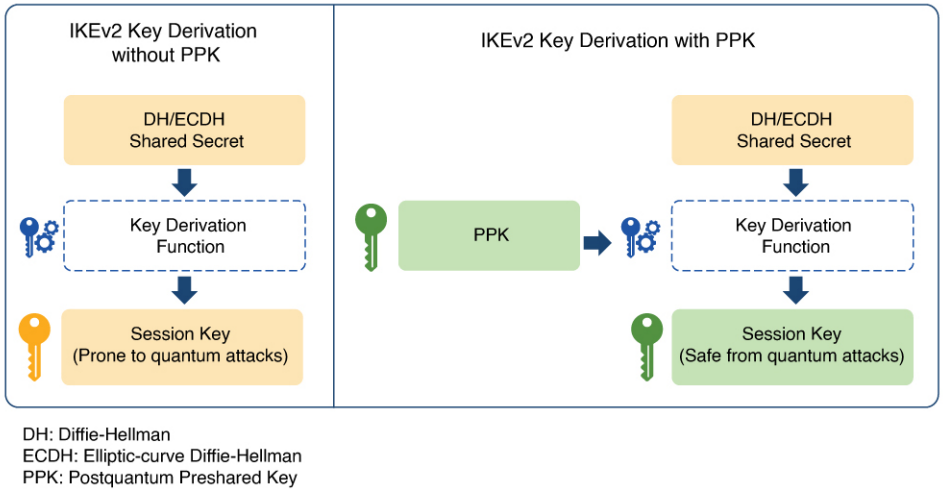
Postquantum Preshared Keys

Session keys that are based on preshared keys are not vulnerable to quantum attacks if the preshared keys have sufficient entropy and the pseudorandom function (PRF), encryption, and authentication transformations are quantum secure. The resulting system is then believed to be secure against classical attackers of today or future attackers with a quantum computer.

RFC 8784 (Mixing Preshared Keys in IKEv2 for Postquantum Security) describes an extension to the IKEv2 protocol to allow it to be resistant to a quantum computer by using preshared keys known as PPKs. The RFC defines negotiation of PPK capability, communication of PPK ID, mixing of PPK as an additional input in the session key derivation, and optional fallback to non-PPK-based session.

Figure 1 shows the IKEv2 key derivation with and without PPK.

Figure 102: IKEv2 Key Derivation - With and Without PPK



DH: Diffie-Hellman
ECDH: Elliptic-curve Diffie-Hellman
PPK: Postquantum Preshared Key

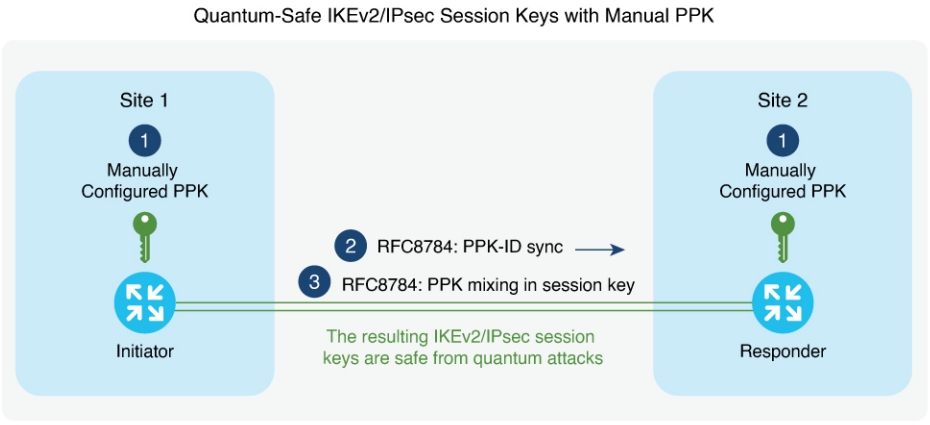
Manual Postquantum Preshared Keys

The simplest provisioning mechanism to provide the same PPKs on the IKEv2 and IPsec initiator and responder pair is to manually configure the PPKs on both sides. The PPKs configured manually are known as manual PPKs.

With a manual PPK, the administrator must ensure that the PPK is of sufficient size and entropy and it is rotated frequently.

Figure 2 shows quantum-safe IKEv2 and IPsec session keys with a manual PPK.

Figure 103: Quantum-Safe IKEv2 and IPsec Session Keys with Manual PPK



Cisco Secure Key Integration Protocol and Dynamic Postquantum Preshared Keys

Cisco SKIP is an HTTPS-based protocol that allows encryption devices such as routers, to import PPKs from an external key source. The externally imported PPKs known as dynamic PPKs offer the benefits of automated provisioning and refresh and better entropy of PPKs.

Cisco SKIP uses TLS1.2 with PSK-DHE cipher suite to make the SKIP protocol quantum-safe. The encryption devices must implement the SKIP client and the external key source must implement the SKIP server.

For an external key source to be SKIP compliant, it must implement the Cisco SKIP protocol and must use an out-of-band synchronization mechanism to provide the same PPK to the two encryption devices—initiator and responder. The external key source can be a Quantum Key Distribution (QKD) device, software, or cloud-based key source or service.

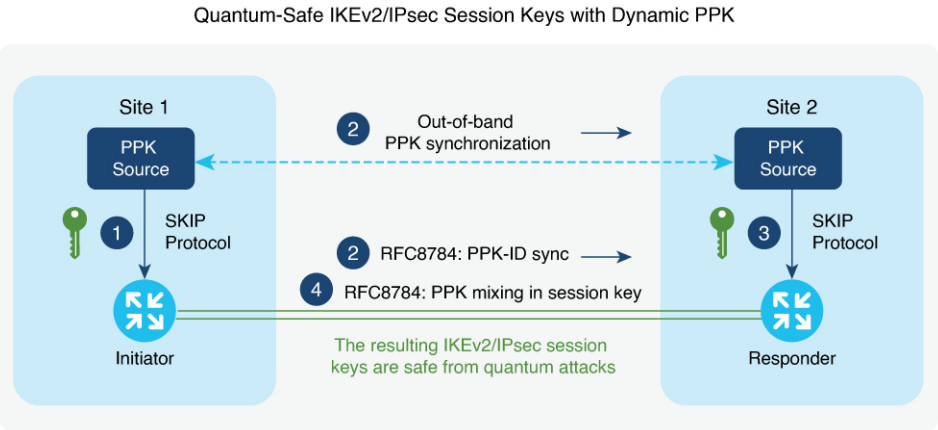
The external key source must meet the following expectations to be SKIP-compliant:

- Must implement SKIP protocol or API, as defined in the Cisco SKIP specification.
- Must provide the same PPK to the encryption device pair—initiator and responder—using an out-of-band synchronization mechanism.

Note Key source vendors, such as QKD vendors, should contact their Cisco representative to implement the Cisco SKIP protocol.

Figure 3 shows quantum-safe IKEv2 and IPsec session keys using dynamic PPK.

Figure 104: Quantum-Safe IKEv2 and IPsec Session Keys with Dynamic PPK



The IKEv2 initiator and responder are connected to their local key source and configured with the SKIP client that specifies the IP address and port of the key source and the preshared key for the TLS1.2 session. The PPK sources are configured with the SKIP parameters, including the local key source identity and the list of identities of the peer key sources.

The following is a high-level operation of the Cisco SKIP protocol:

1. The IKEv2 initiator places a request for a PPK from its key source. The key source replies with a PPK and the corresponding PPK ID.
2. The initiator-side key source synchronizes the PPK to the responder-side key source using an out-of-band mechanism that is specific to the type of key source. The IKEv2 initiator communicates the PPK ID to the IKEv2 responder over IKEv2 using the RFC 8784 extensions.
3. The IKEv2 responder requests from its key source, the PPK corresponding to the PPK ID received from the IKEv2 initiator. The key source replies with the PPK corresponding to the PPK ID.
4. The IKEv2 initiator and responder mix the PPK in the key derivation, as specified in RFC 8784. The resulting IKEv2 and IPsec session keys are quantum-safe.

How to Configure Quantum-Safe Encryption Using Postquantum Preshared Keys

The following sections describe the processes involved in configuring quantum-safe encryption using postquantum preshared keys.

Configuring Manual Postquantum Preshared Keys

Perform the following tasks to configure the manual PPK.

Configuring Manual Postquantum Preshared Keys in an IKEv2 Keyring

Follow these steps to configure the manual PPK for one or more peers or groups of peers, in the IKEv2 keyring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 keyring** *keyring-name*
4. **peer** *name*
5. Run one of the following commands:
   - **address** {*ipv4-address mask* | *ipv6-address prefix*}
   - **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
6. **ppk manual id** *ppk-id* **key** [**0** | **6** | **hex**] *password* [**required**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 keyring** *keyring-name* **Example:** `Device(config)# crypto ikev2 keyring keyring1`	Defines an IKEv2 keyring and enters IKEv2 keyring configuration mode.
Step 4	**peer** *name* **Example:** `Device(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group and enters IKEv2 keyring peer configuration mode.
Step 5	Run one of the following commands: • **address** {*ipv4-address mask* | *ipv6-address prefix*} • **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*} **Example:**	Specifies the remote IKEv2 peers based on WAN IP address or IKEv2 identity. • The **address** command specifies an IPv4 or IPv6 address or range for the peer or group of peers. **Note** This IP address is the IKE endpoint address and is independent of the identity address.

	Command or Action	Purpose

	`Device(config-ikev2-keyring-peer)# address 10.0.0.1 255.0.0.0`
	Example:
	`Device(config-ikev2-keyring-peer)# identity address 10.0.0.1`
		• The **identity** command identifies the IKEv2 peer through the following identities:
		• E-mail
		• Fully qualified domain name (FQDN)
		• IPv4 or IPv6 address
		• Key ID
		Note The **identity** command is available for key lookup only on the IKEv2 responder.

Step 6	**ppk manual id** *ppk-id* **key** [**0** | **6** | **hex**] *password* [**required**] **Example:** `Device(config-ikev2-keyring-peer)# ppk manual id ppk_id key cisco123`	Configures PPK ID and PPK for the identified peers. • **ppk manual**: Indicates that the PPK ID and the PPK are configured manually. • **id** *ppk-id*: Specifies the PPK ID. • **key** *password*: Specifies the PPK. • **required**: Indicates that the quantum-safe encryption using PPK is mandatory and there must be no fallback to a normal IKEv2 or IPsec session. **Note** The *ppk-id* and the PPK must match on both the peers.

Configuring an IKEv2 Keyring in an IKEv2 Profile

SUMMARY STEPS

1. **crypto ikev2 profile** *profile-name*
2. **keyring ppk** *keyring-name*
3. **exit**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ikev2 profile** *profile-name* **Example:** `Device(config-ikev2-keyring-peer)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 2	**keyring ppk** *keyring-name* **Example:**	Specifies the keyring that has either manual or dynamic PPK configured.

	Command or Action	Purpose
	`Device(config-ikev2-profile)# keyring ppk keyring1`	**Note** To remove the keyring from the IKEv2 profile, use the **no keyring** {**aaa** | **local** | **ppk**} *keyring-name* command.
Step 3	**exit** **Example:** `Device(config-ikev2-profile)# exit`	Exits IKEv2 profile configuration mode and returns to global configuration mode.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring Dynamic Postquantum Preshared Keys

Perform the following tasks to configure the dynamic PPK.

Configuring a Secure Key Integration Protocol Client

SKIP client configuration specifies the parameters required to securely communicate with and request PPKs from an external SKIP-compliant key source.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto skip-client** *skip-client-name*
4. **server** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address* | **fqdn** *domain-name*} **port** *port-number*
5. **psk id** *id-name* **key** [**0** | **6** | **hex**] *password*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto skip-client** *skip-client-name* **Example:** `Device(config-crypto-skip-client)# crypto skip-client skip-client-cfg`	Specifies the name of SKIP client configuration block and enters SKIP client configuration mode.

	Command or Action	Purpose
Step 4	**server** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address* | **fqdn** *domain-name*} **port** *port-number* **Example:** `Device(config-crypto-skip-client)# server ipv4 10.10.0.3 port 9993`	Specifies the IP address or FQDN and port to connect to the external key source.
Step 5	**psk id** *id-name* **key** [**0** | **6** | **hex**] *password* **Example:** `Device(config-crypto-skip-client)# psk id psk-id key 0 cisco123`	Specifies the preshared key identity and the preshared key for the SKIP TLS session.
Step 6	**exit** **Example:** `Device(config-crypto-skip-client)# exit`	Exits SKIP client configuration mode and returns to global configuration mode.

Configuring a Secure Key Integration Protocol Client in an IKEv2 Keyring

Follow these steps to configure the manual PPK for one or more peers or groups of peers in the IKEv2 keyring.

SUMMARY STEPS

1. **crypto ikev2 keyring** *keyring-name*
2. **peer** *name*
3. Execute one of the following commands:
   - **address** {*ipv4-address mask* | *ipv6-address prefix*}
   - **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
4. **ppk dynamic** *skip-client-name* [**required**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ikev2 keyring** *keyring-name* **Example:** `Device(config)# crypto ikev2 keyring keyring1`	Defines an IKEv2 keyring and enters IKEv2 keyring configuration mode.
Step 2	**peer** *name* **Example:** `Device(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group and enters IKEv2 keyring peer configuration mode.
Step 3	Execute one of the following commands: • **address** {*ipv4-address mask* | *ipv6-address prefix*} • **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}	Specifies the remote IKEv2 peers based on WAN IP address or IKEv2 identity. The **address** command specifies an IPv4 or IPv6 address or range for the peer or group of peers.

	Command or Action	Purpose

	Example:
	`Device(config-ikev2-keyring-peer)# address 10.0.0.1 255.0.0.0`
	Example:
	`Device(config-ikev2-keyring-peer)# identity address 10.0.0.1`
		Note This IP address is the IKE endpoint address and is independent of the identity address.
		The **identity** command identifies the IKEv2 peer through the following identities:
		• E-mail
		• Fully qualified domain name (FQDN)
		• IPv4 or IPv6 address
		• Key ID
		Note The **identity** command is available for key lookup only on the IKEv2 responder.

Step 4	**ppk dynamic** *skip-client-name* [**required**] **Example:** `Device(config-ikev2-keyring-peer)# ppk dynamic skip-client1`	Specifies the external key source to use for dynamic PPKs. • **ppk dynamic**: Indicates that the PPK is imported dynamically from an external key source. • **required**: Indicates that the quantum-safe encryption using PPK is mandatory and there must be no fallback to a normal IKEv2 or IPsec session.

Configuring an IKEv2 Keyring in an IKEv2 Profile

SUMMARY STEPS

1. **crypto ikev2 profile** *profile-name*
2. **keyring ppk** *keyring-name*
3. **exit**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ikev2 profile** *profile-name* **Example:** `Device(config-ikev2-keyring-peer)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 2	**keyring ppk** *keyring-name* **Example:** `Device(config-ikev2-profile)# keyring ppk keyring1`	Specify the keyring that has either manual or dynamic PPK configured. **Note** To remove the keyring from the IKEv2 profile, use the **no keyring** {**aaa** | **local** | **ppk**} *keyring-name* command.

	Command or Action	Purpose
Step 3	**exit** **Example:** `Device(config-ikev2-profile)# exit`	Exits IKEv2profile configuration mode and returns to global configuration mode.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Quantum-Safe Encryption Using Postquantum Preshared Keys

The following sections provide detailed configuration examples relating to the configuration of quantum-safe encryption using PPKs.

Example: Configuring the Manual Postquantum Preshared Keys

Example: Initiator Configuration

The following example shows how to manually configure a PPK for an initiator:

```
conf t
hostname Router1
!
crypto ikev2 keyring ppk-keyring
peer 1
address 10.10.0.1 255.255.255.0
ppk manual id ppk_id key cisco123
!
crypto ikev2 profile prof
match identity remote address 10.10.0.1
authentication local pre-share key cisco
authentication remote pre-share key cisco
keyring ppk ppk-keyring
!
crypto ipsec profile prof
set ikev2-profile prof
!
interface Tunnel0
ip address 10.10.0.1 255.255.255.0
tunnel source GigabitEthernet1
tunnel destination 10.10.10.1
tunnel protection ipsec profile prof
!
interface GigabitEthernet1
ip address 10.10.10.2 255.255.255.0
no shut
!
```

Example: Responder Configuration

The following example shows how to manually configure a PPK for a responder:

```
conf t
hostname Router2
!
crypto ikev2 keyring ppk-keyring
peer 1
address 10.10.0.1 255.255.255.0
ppk manual id ppk_id key cisco
!
crypto ikev2 profile prof
match identity remote address 10.10.0.1
authentication local pre-share key cisco
authentication remote pre-share key cisco
keyring ppk ppk-keyring
!
crypto ipsec profile prof
set ikev2-profile prof
!
interface Tunnel0
ip address 10.10.0.2 255.255.255.0
tunnel source GigabitEthernet1
tunnel destination 10.10.10.2
tunnel protection ipsec profile prof
!
interface GigabitEthernet1
ip address 10.10.0.1 255.255.255.0
no shut
!
```

Example: Configuring the Dynamic Postquantum Preshared Keys

Example: Initiator Configuration

The following example shows how to configure a dynamic PPK for an initiator:

```
conf t
hostname Router1
!
crypto skip-client skip-client-cfg
server ipv4 10.10.0.4  port 9991
psk id psk-id1 key 0 cisco123
!
crypto ikev2 keyring ppk-keyring
peer 1
address 10.10.0.1 255.255.255.0
ppk dynamic skip-client-cfg
!
crypto ikev2 profile prof
match identity remote address 10.10.0.1
authentication local pre-share key cisco
authentication remote pre-share key cisco
keyring ppk ppk-keyring
!
crypto ipsec profile prof
set ikev2-profile prof
!
interface Tunnel0
ip address 10.10.0.2 255.255.255.0
```

```
tunnel source GigabitEthernet1
tunnel destination 10.10.10.1
tunnel protection ipsec profile prof
!
interface GigabitEthernet1
ip address 10.10.10.2 255.255.255.0
no shut
!
interface GigabitEthernet1
ip address 10.10.10.3 255.255.255.0
no shut
!
```

Example: Responder Configuration

The following example shows how to configure a dynamic PPK for a responder:

```
conf t
hostname Router2
!
crypto skip-client skip-client-cfg
server ipv4 10.10.0.4 port 9992
psk id vedge-sim-1 key 0 cisco123
!
crypto ikev2 keyring ppk-keyring
peer 1
address 10.10.0.1 255.255.255.0
ppk dynamic skip-client-cfg
!
crypto ikev2 profile prof
match identity remote address 10.10.0.1
authentication local pre-share key cisco
authentication remote pre-share key cisco
keyring ppk ppk-keyring
!
crypto ipsec profile prof
set ikev2-profile prof
!
interface Tunnel0
ip address 10.10.0.2 255.255.255.0
tunnel source GigabitEthernet1
tunnel destination 10.10.10.2
tunnel protection ipsec profile prof
!
interface GigabitEthernet1
ip address 10.10.10.1 255.255.255.0
no shut
!
interface GigabitEthernet1
ip address 10.10.10.4 255.255.255.0
!
```

Verifying the Postquantum Preshared Keys Configuration

Use the **show crypto ikev2 sa detailed** command to display information about the current IKEv2 security associations. The `Quantum Resistance Enabled` message displayed in the output indicates that PPK-based quantum-safe encryption is enabled.

The following is a sample output from the **show crypto ikev2 sa detailed** command:

```
IPv4 Crypto IKEv2  SA
Tunnel-id       Local               Remote                  fvrf/ivrf          Status

   3        <src IP>/SrcPort     <Dst IP>/DstPort             none/none        READY

Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19,
Auth sign:
.
.
.
Initiator of SA : No
Quantum Resistance Enabled
```

Additional References for Quantum-Safe Encryption Using Postquantum Preshared Keys

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	Configuring Security for VPNs with IPsec

RFCs

RFC	Title
RFC 8784	*Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Postquantum Security*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Quantum-Safe Encryption Using Postquantum Preshared Keys

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 272: Feature Information for Quantum-Safe Encryption Using Postquantum Preshared Keys

Feature Name	Releases	Feature Information
Quantum-Safe Encryption Using Postquantum Preshared Keys	Cisco IOS XE Release 17.11.1a	The feature implements RFC 8784 and Cisco Secure Key Integration Protocol (SKIP) for quantum-safe encryption of IKEv2 and IPsec packets using Postquantum Preshared Keys (PPKs). The PPKs that are configured manually are known as manual PPKs, and the PPKs that are imported from an external key source using the SKIP protocol are known as dynamic PPKs.
Quantum-Safe Encryption Using Postquantum Preshared Keys	Cisco IOS XE Release 17.12.1a	This enhancement introduces support for Quantum-Safe Encryption Using Postquantum Preshared Keys for the following platforms: • Cisco 1000 Series Integrated Services Routers • Cisco Catalyst 8500 Series Edge Platforms

CHAPTER 207

Configuring the FlexVPN Server

This module describes FlexVPN server features, IKEv2 commands required to configure the FlexVPN server, remote access clients, and the supported RADIUS attributes.

Note Security threats, as well as cryptographic technologies to help protect against such threats, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Restrictions for the FlexVPN Server

Dual-Stack Tunnel Interface and VRF-Aware IPsec

When configuring a dual-stack tunnel interface in a VPN routing and forwarding (VRF)-aware IPsec scenario, you cannot use the **ip vrf forwarding** command to configure an Inside VPN routing and forwarding (IVRF) instance because this is not a valid configuration. Use the **vrf forwarding** *vrf-name* command to define the IVRF of the tunnel interface, where the *vrf-name* argument is defined using the **vrf definition** command with IPv4 and IPv6 address families inside the definition.

SSO Restrictions

- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.

Information About the FlexVPN Server

Peer Authentication Using EAP

The FlexVPN server supports peer authentication using the Extensible Authentication protocol (EAP) and acts as a pass-through authenticator relaying EAP messages between the client and the backend EAP server. The backend EAP server is typically a RADIUS server that supports EAP authentication.

Note While a FlexVPN client authenticates the FlexVPN client using EAP, the FlexVPN server must authenticated by using certificates.

The FlexVPN server is configured to authenticate FlexVPN clients that use EAP by configuring the **authentication remote eap** command in IKEv2 profile configuration mode. FlexVPN clients authenticate using EAP by skipping the AUTH payload in the IKE_AUTH request.

If the **query-identity** keyword is configured, the FlexVPN server queries the EAP identity from the client; otherwise, the FlexVPN client's IKEv2 identity is used as the EAP identity. However, if the **query-identity** keyword is not configured and the FlexVPN client's IKEv2 identity is an IPv4 or IPv6 address, the session is terminated because IP addresses cannot be used as the EAP identity.

The FlexVPN server starts the EAP authentication by passing the FlexVPN client's EAP identity to the EAP server; the FlexVPN server then relays EAP messages between the remote access (RA) client and the EAP server until the authentication is complete. If the authentication succeeds, the EAP server is expected to return the authenticated EAP identity to the FlexVPN server in the EAP success message.

After EAP authentication, the EAP identity used for the IKEv2 configuration is obtained from the following sources in the given order:

- The EAP identity provided by the EAP server with the EAP success message.
- The EAP identity queried from the client when the **query-identity** keyword is configured.
- The FlexVPN client IKEv2 identity used as the EAP identity.

The figure below shows IKEv2 exchange for EAP authentication without the **query-identity** keyword.

Figure 105: IKEv2 Exchange Without the query-identity Keyword

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	RADIUS Access-Request/ EAP-Message/EAP-Response/(EAP-ID(IKEv2-ID)) →	
		← RADIUS Access-Challenge/EAP-Message/ EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method)} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (other attributes))
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

209140

The figure below shows the IKEv2 exchange for EAP authentication with the **query-identity** keyword.

Figure 106: IKEv2 Exchange with the query-identity Keyword

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	← HDR, SK {IDr, [CERT,] AUTH, EAP (EAP-request (Identity)) }	
HDR, SK {EAP(EAP-Response(Identity)} →		
	RADIUS Access-Request/ EAP-Message/EAP-Response/(EAP-ID) →	
		← RADIUS Access-Challenge/EAP-Message/ EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method)} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (EAP-identity) (other attributes))
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

209141

IKEv2 Configuration Mode

IKEv2 configuration mode allows IKE peers to exchange configuration information such as IP addresses and routes. The configuration information is obtained from IKEv2 authorization. Both pull and push models are supported. The pull model involves the exchange of configuration requests and replies; the push model involves the exchange of configuration sets and acknowledgements.

The following table describes the conditions when the initiator and the responder send different configuration payload types:

Table 273: Configuration Payload Types

Configuration Payload Type	Sent By...	When...
CFG_REQUEST	Initiator	The initiator is the FlexVPN client or if the **config-exchange request** command is enabled in the IKEv2 profile.
CFG_REPLY	Responder	The responder receives the CFG_REQUEST.
CFG_SET	Initiator and responder	Initiator—The **config-exchange set send** command is enabled in the IKEv2 profile. Responder—The CFG_REQUEST is not received, the configuration data is available, and the **config-exchange set send** command is enabled in the IKEv2 profile.
CFG_ACK	Initiator and responder	Initiator—The **config-exchange set accept** command is enabled in the IKEv2 profile. Responder—The **config-exchange set accept** command is enabled in the IKEv2 profile.

Note The commands to send configuration requests and configuration set payloads are enabled by default.

Depending on your release, the IKEv2 initiator can trigger a configuration mode when the initiator is a FlexVPN client, or any static tunnel interface initiating IKEv2 can trigger configuration mode by enabling the **config-mode** command in the IKEv2 profile.

The IKEv2 FlexVPN server supports the following standard IPv4 configuration attributes:

- INTERNAL_IP4_ADDRESS
- INTERNAL_IP4_NETMASK
- INTERNAL_IP4_DNS
- INTERNAL_IP4_NBNS
- INTERNAL_IP4_SUBNET

The IKEv2 FlexVPN server supports the following standard IPv6 configuration attributes:

- INTERNAL_IP6_ADDRESS
- INTERNAL_IP6_DNS
- INTERNAL_IP6_SUBNET

Note IPv6 configuration attributes are only supported by the Microsoft Windows IKEv2 client.

The INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET configuration attributes, controlled by the **route set** and **aaa attribute list** commands in the IKEv2 authorization policy, are not supported when you configure a static virtual tunnel interface (SVTI)-to-SVTI tunnel. In such cases, static routing or dynamic routing must be used instead of the IKEv2-based route exchange.

The IKEv2 FlexVPN server supports the following standard common configuration attribute:

- APPLICATION_VERSION

Note This attribute is only sent for Cisco Anyconnect and FlexVPN clients.

The IKEv2 FlexVPN server supports the following Cisco Unity configuration attributes:

- MODECFG_BANNER
- MODECFG_DEFDOMAIN
- MODECFG_SPLITDNS_NAME
- MODECFG_BACKUPSERVERS
- MODECFG_PFS
- MODECFG_SMARTCARD_REMOVAL_DISCONNECT

Note The Cisco Unity attributes are sent only for Cisco Anyconnect and FlexVPN clients.

The IKEv2 FlexVPN server supports the following Cisco FlexVPN configuration attributes:

- MODECFG_CONFIG_URL
- MODECFG_CONFIG_VERSION

Note The Cisco FlexVPN attributes are sent only for Cisco FlexVPN clients.

The INTERNAL_IP4_ADDRESS attribute value is derived from the following sources in the given order:

- The Framed-IP-Address attribute received in AAA user authorization.
- The local IP address pool.
- The DHCP server.

The DHCP server, if configured, allocates addresses only if the local IP address pool is not configured. However, if an error occurs when allocating IP addresses from the local pool, the next address source DHCP server is not used for allocating the addresses.

The value for INTERNAL_IP4_NETMASK attribute is derived as follows:

- If the IP address is obtained from the DHCP server, the netmask is also obtained from the DHCP server.

- If the IP address is obtained from either the Framed-IP-Address attribute in AAA user authorization or the local IP address pool, the netmask is derived from the IPv4 netmask attribute received in the user or group authorization. If the netmask is not available, the INTERNAL_IP4_NETMASK attribute is not included in the configuration reply. If the netmask is available, the INTERNAL_IP4_NETMASK attribute is included only if the INTERNAL_IP4_ADDRESS attribute is included in the configuration reply.

An IPv4 address is allocated and included in the reply only if the client requests an address. If the client requests multiple IPv4 addresses, only one IPv4 address is sent in the reply. If available, the remaining attributes are included in the reply even though the client does not request them. If the client requests an IPv4 address and the FlexVPN server is unable to assign an address, an INTERNAL_ADDRESS_FAILURE message is returned to the client.

It is always recommended that the prefix length should be used as 128 on ipv6 local pool configuration.

For example, if clients are 4 , **ipv6 local pool pool1 afe0::/126 128** needed to be configured for the prefix length. If clients are 16, **ipv6 local pool pool1 afe0::/124 128** needed to be configured for the prefix length.

IKEv2 Authorization

IKEv2 authorization provides a policy for an authenticated session by using the AAA. The policy can be defined locally or on the RADIUS server, and contains local and/or remote attributes. The username for authorization can either be derived from the peer identity using the **name-mangler** keyword or be directly specified in the command. IKEv2 authorization is mandatory only if the peer requests an IP address via configuration mode.

IKEv2 authorization types are as follows:

- User authorization—Use the **aaa authorization user** command in the IKEv2 profile to enable user authorization. User authorization is based on the user-specific portion of the peer IKE identity such as fqdn-hostname. The attributes from user authorization are called user attributes.
- Group authorization—Use the **aaa authorization group** command in the IKEv2 profile to enable group authorization. Group authorization is based on the generic portion of the peer IKE identity such as fqdn-domain. The attributes from group authorization are called group attributes.
- Implicit user authorization—Use the **aaa authorization user cached** command in the IKEv2 profile to enable implicit user authorization. Implicit authorization is performed as part of EAP authentication or when obtaining the AAA preshared key. The attributes from implicit user authorization are called cached attributes.

Note Depending on your release, the **aaa authorization user cached** command may or may not be available. Explicit user authorization is performed only when implicit user authorization does not return any attributes or does not have the Framed-IP-Address attribute.

Merging and Overriding Attributes

Attributes from different sources are merged before they are used. The precedence of merging attributes is as follows:

- When merging duplicate attributes, the source of the attribute has a higher precedence.
- When merging user and cached attributes, user attributes have higher precedence.

- When merging merged-user-attributes and group attributes, merged-user attributes have a higher precedence, by default. However, this precedence can be reversed using the **aaa author group override** command.

IKEv2 Authorization Policy

An IKEv2 authorization policy defines the local authorization policy and contains local and/or remote attributes. Local attributes, such as VPN routing and forwarding (VRF) and the QOS policy, are applied locally. Remote attributes, such as routes, are pushed to the peer via the configuration mode. Use the **crypto ikev2 authorization policy** command to define the local policy. The IKEv2 authorization policy is referred from the IKEv2 profile via the **aaa authorization** command.

IKEv2 Name Mangler

The IKEv2 name mangler is used to derive the username for IKEv2 authorization and obtain the AAA preshared key from the peer IKE identity.

IKEv2 Multi-SA

The IKEv2 Multi-SA feature allows an IKEv2 Dynamic Virtual Tunnel Interface (DVTI) session on the IKEv2 responder to support multiple IPsec Security Associations (SA). The maximum number of IPsec SAs per DVTI session is either obtained from AAA authorization or configured on the IPsec profile. The value from AAA has a higher priority. Any change to the *max-flow-limit* argument in the IPsec profile is not applied to the current session but is applied to subsequent sessions. The IKEv2 Multi-SA feature makes the configuration of the IKEv2 profile in the IPsec profile optional. This optional configuration allows IPsec DVTI sessions using the same virtual template to have different IKEv2 profiles, thus saving the number of virtual template configurations.

Note The IKEv2 Multi-SA feature allows multiple IPsec SAs that have non-any-any proxies. However, when the IPsec SA proxies are any-any, a single IPsec SA is allowed.

For more information, see the "Multi-SA Support for Dynamic Virtual Tunnel Interfaces for IKEv2" module in the *Security for VPNs with IPsec Configuration Guide*.

AnyConnect Profile Download

The FlexVPN AnyConnect Profile Download feature enables a device running Cisco IOS XE software to connect and push the profile information in IKEv2 protocol to Cisco AnyConnect Secure Mobility client.

Cisco AnyConnect Secure Mobility client contains profiles used for configuring the settings for VPN. These profiles can be manually configured or can be downloaded from a headend. The headend can be configured to deploy profiles globally for all Cisco AnyConnect Secure Mobility client users.

Use the **anyconnect profile** command to match a VPN profile against an IKEv2 profile.

Note Crypto SSL profile is not mandatory to configure AnyConnect Profile Download feature.

Supported RADIUS Attributes

The following tables list the RADIUS attributes supported by the IKEv2 FlexVPN server:

- The Scope field defines the direction of the attribute and the usage on the FlexVPN server or client.
  - Inbound—FlexVPN server to RADIUS
  - Outbound—RADIUS to the FlexVPN server
  - Local—Used locally by the FlexVPN server
  - Remote—Pushed to the client by the FlexVPN server
- The "Local configuration" field specifies the IKEv2 authorization policy command that is used to configure the attribute locally on the FlexVPN server.
- Cisco AV Pair is a Cisco Vendor Specific Attribute (VSA) with vendor-id 9 and vendor-type 1. The VSAs are encapsulated in the Radius IETF attribute 26 Vendor-Specific. The Cisco AV pair is specified as a string of format "protocol:attribute=value".

Example:

```
cisco-avpair = "ipsec:ipv6-addr-pool=v6-pool"
```

The following example shows the Cisco AV pair for a standard access-list.

```
cisco-avpair = "ipsec:route-set=access-list 99"
```

Table 274: Inbound and Bidirectional IETF RADIUS Attributes

Attribute	Scope
User-Name	Inbound and outbound (bidirectional)
User-Password	Inbound
Calling-Station-Id	Inbound
Service-Type	Inbound
EAP-Message	Bidirectional
Message-Authenticator	Bidirectional

Table 275: Outbound IETF and Cisco AV Pair RADIUS Attributes

Attribute	Type	Scope	Local configuration
Tunnel-Type	IETF	Local	N/A
Tunnel-Medium-Type	IETF	Local	N/A

Attribute	Type	Scope	Local configuration
Tunnel-Password	IETF	Local	N/A
ipsec:ikev2-password-local	Cisco AV Pair	Local	N/A
ipsec:ikev2-password-remote	Cisco AV Pair	Local	N/A
ipsec:addr-pool	Cisco AV Pair	Local	pool
ipsec:group-dhcp-server	Cisco AV Pair	Local	dhcp server
ipsec:dhcp-giaddr	Cisco AV Pair	Local	dhcp giaddr
ipsec:dhcp-timeout	Cisco AV Pair	Local	dhcp timeout
ipsec:ipv6-addr-pool	Cisco AV Pair	Local	ipv6 pool
ipsec:route-set=interface	Cisco AV Pair	Local	route set interface
ipsec:route-set=prefix	Cisco AV Pair	Local	N/A
ipsec:route-accept	Cisco AV Pair	Local	route accept any
ip:interface-config	Cisco AV Pair	Local	aaa attribute list
ipsec:ipsec-flow-limit	Cisco AV Pair	Local	ipsec flow-limit
Framed-IP-Address	IETF	Remote	N/A
Framed-IP-Netmask	IETF	Remote	netmask
ipsec:dns-servers	Cisco AV Pair	Remote	DNS
ipsec:wins-servers	Cisco AV Pair	Remote	wins
ipsec:route-set=access-list (See Note 1.)	Cisco AV Pair	Remote	route set access-list (See Note 1.)
ipsec:addrv6	Cisco AV Pair	Remote	n/a
ipsec:prefix-len	Cisco AV Pair	Remote	n/a
ipsec:ipv6-dns-servers-addr	Cisco AV Pair	Remote	ipv6 dns
ipsec:route-set=access-list ipv6	Cisco AV Pair	Remote	route set access-list ipv6
ipsec:banner	Cisco AV Pair	Remote	banner
ipsec:default-domain	Cisco AV Pair	Remote	def-domain
ipsec:split-dns	Cisco AV Pair	Remote	split-dns
ipsec:ipsec-backup-gateway	Cisco AV Pair	Remote	backup-gateway
ipsec:pfs	Cisco AV Pair	Remote	pfs

Attribute	Type	Scope	Local configuration
ipsec:include-local-lan	Cisco AV Pair	Remote	include-local-lan
ipsec:smartcard-removal-disconnect	Cisco AV Pair	Remote	smartcard-removal- disconnect
ipsec:configuration-url	Cisco AV Pair	Remote	configuration url
ipsec:configuration-version	Cisco AV Pair	Remote	configuration version

Note

- 1. The RADIUS attribute to set an access list on IKEv2 FlexVPN server only supports a standard access list. An extended access list is not supported.

Supported Remote Access Clients

The FlexVPN server interoperates with the Microsoft Windows7 IKEv2 client, Cisco IKEv2 AnyConnect client, and Cisco FlexVPN client.

Microsoft Windows7 IKEv2 Client

The Microsoft Windows 7 IKEv2 client sends an IP address as the Internet Key Exchange (IKE) identity that prevents the Cisco IKEv2 FlexVPN server from segregating remote users based on the IKE identity. To allow the Windows 7 IKEv2 client to send the email address (user@domain) as the IKE identity, apply the hotfix documented in KB975488 (http://support.microsoft.com/kb/975488) on Microsoft Windows 7 and specify the email address string in either the Username field when prompted or the CommonName field in the certificate depending on the authentication method.

For certificate-based authentication, the FlexVPN server and Microsoft Windows 7 client certificates must have an Extended Key Usage (EKU) field as follows:

- For the client certificate, EKU field = client authentication certificate.
- For the server certificate, EKU field = server authentication certificate
- The certificates can be obtained from the Microsoft Certificate Server or the IOS CA server.

For EAP authentication, the Microsoft Windows 7 IKEv2 client expects an EAP identity request before any other EAP requests. Ensure that you configure the **query-identity** keyword in the IKEv2 profile on the IKEv2 FlexVPN server to send an EAP identity request to the client.

Cisco IKEv2 AnyConnect Client

For certificate-based authentication, the FlexVPN server and the AnyConnect client certificates must have an Extended Key Usage (EKU) field as follows:

- For the client certificate, EKU field = client authentication certificate
- For the server certificate, EKU field = server authentication certificate

If the FlexVPN server authenticates to AnyConnect client using certificates, a SubjectAltName extension is required in the FlexVPN server certificate that contains the server's IP address or fully qualified domain name

(FQDN). Additionally, HTTP certified URLs must be disabled on the FlexVPN server using the **no crypto ikev2 http-url cert** command.

The following example displays the XML tags specific to EAP-MD5 authentication of IKEv2 sessions in the AnyConnect client profile:

```
<PrimaryProtocol>IPsec
       <StandardAuthenticationOnly>true
           <AuthMethodDuringIKENegotiation>
               EAP-MD5
           </AuthMethodDuringIKENegotiation>
           <IKEIdentity>DEPT24</IKEIdentity>
       </StandardAuthenticationOnly>
</PrimaryProtocol>
```

Note For every flap or FlexVPN tunnel that is enabled, the following message is displayed:

```
*Jan 22 22:52:09.833: %SYS-5-CONFIG_P: Configured programmatically by process Crypto INT
from console as console
*Jan 22 22:52:09.840: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2,
changed state to up
```

For more information, refer to AnyConnect client 3.0 documentation at this link: http://www.cisco.com/en/US/docs/security/vpn_client/anyconnect/anyconnect30/release/notes/anyconnect30rn.html#wp1268255.

How to Configure the FlexVPN Server

Configuring the IKEv2 Profile for the FlexVPN Server

This task describes the IKEv2 profile commands required for configuring the FlexVPN server in addition to the basic IKEv2 profile commands. Refer to the "Configuring IKEv2 Profile (Basic)" task in the *Configuring Internet Key Exchange Version 2 (IKEv2)* feature module for information about configuring the basic IKEv2 profile.

Perform this task to configure the IKEv2 profile for the FlexVPN Server:

Step 1 **enable**

Example:

Enables privileged EXEC mode.

```
Device> enable
```

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

Enters the global configuration mode.

```
Device# configure terminal
```

Step 3 **crypto ikev2 profile** *profile-name*

Defines an IKEv2 profile name and enters IKEv2 profile configuration mode.

Example:

```
Device(config)# crypto ikev2 profile profile1
```

Step 4 **aaa authentication eap** *list-name*

Example:

```
Device(config-ikev2-profile)# aaa authentication eap list1
```

(Optional) Specifies the AAA authentication list for the EAP authentication when implementing the IKEv2 remote access server.

- **eap**—Specifies the external EAP server.
- *list-name*—The AAA authentication list name.

Step 5 **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}

Example:

```
Device(config-ikev2-profile)# authentication local ecdsa-sig
```

Specifies the local or remote authentication method.

- **rsa-sig**—Specifies RSA-sig as the authentication method.
- **pre-share**—Specifies the preshared key as the authentication method.
- **ecdsa-sig**—Specifies ECDSA-sig as the authentication method.
- **eap**—Specifies EAP as the remote authentication method.
- **query-identity**—Queries the EAP identity from the peer.
- **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response.

Note You can specify only one local authentication method but multiple remote authentication methods.

Step 6 Execute both or one of the following:

- **aaa authorization user** {**eap** | **psk**} {**cached** | **list** *aaa-listname* [*aaa-username* | **name-mangler** *mangler-name*]}
- **aaa authorization user cert list** *aaa-listname* {*aaa-username* | **name-mangler** *mangler-name*}

Example:

```
Device(config-ikev2-profile)# aaa authorization user eap cached
```

Example:

```
Device(config-ikev2-profile)# aaa authorization user cert list list1 name-mangler mangler1
```

Specifies the AAA method list and username for user authorization.

- **user**—Specifies user authorization.
- **cert**—Specifies that the peers must be authenticated using certificates.

- **eap**—Specifies that the peers must be authenticated using EAP.
- **psk**—Specifies that the peers must be authenticated using preshared keys.
- **cached**—Specifies that the attributes received during EAP authentication or obtained from the AAA preshared key must be cached.
- *aaa-listname*—AAA method list name.
- *aaa-username*—Specifies the username that must be used in the AAA authorization request.
- **name-mangler**—Specifies the name mangler that derives the AAA authorization username from the peer identity.
- *mangler-name*—Name mangler to be used.

Note
- For **psk** and **eap** authentication methods, specifying the *aaa-username* argument or the **name-mangler** keyword is optional and if not specified, the peer identity is used as the username.
- For **psk** and **eap** authentication methods, you can simultaneously configure two variants for user authorization with the **cached** and **list** keyword respectively.
- Specifying the *aaa-username* argument or the **name-mangler** keyword is mandatory for **cert** authentication, as the peer identity of type distinguished name (DN) cannot be used.

Step 7 Execute both or one of the following:

- **aaa authorization group** [**override**] {**eap** | **psk**} **list** *aaa-listname* [*aaa-username* | **name-mangler** *mangler-name*]
- **aaa authorization group** [**override**] **cert list** *aaa-listname* {*aaa-username* | **name-mangler** *mangler-name*}

Example:

```
Device(config-ikev2-profile)# aaa authorization group override psk list list1
```

Example:

```
Device(config-ikev2-profile)# aaa authorization group cert list list1 name-mangler mangler1
```

Specifies the AAA method list and username for group authorization.

- **group**—Specifies group authorization.
- **override**—(Optional) Specifies that attributes from group authorization should take precedence while merging attributes. By default, user attributes take precedence.
- **cert**—Specifies that peers must be authenticated using certificates.
- **eap**—Specifies that peers must be authenticated using EAP.
- **psk**—Specifies that peers must be authenticated using preshared keys.
- *aaa-listname*—AAA method list name.
- *aaa-username*—Username that must be used in the AAA authorization request.
- **name-mangler**—Specifies the name mangler that derives the AAA authorization username from the peer identity.
- *mangler-name*—Name mangler to be used.

Note
- For **psk** and **eap** authentication methods, specifying the *aaa-username* argument or the **name-mangler** keyword is optional and if not specified, the peer identity is used as the username.
- For **psk** and **eap** authentication methods, you can simultaneously configure two variants for user authorization with the **cached** and **list** keyword respectively.
- Specifying the *aaa-username* argument or the **name-mangler** keyword is mandatory for **cert** authentication, as the peer identity of type distinguished name (DN) cannot be used.

Step 8 **config-exchange** {**request** | **set** {**accept** | **send**}}

Example:

```
Device(config-ikev2-profile)# config-exchange set accept
```

(Optional) Enables configuration exchange options.

- **request**—Enables the configuration exchange request.
- **set**—Enables the configuration exchange request set options.
- **accept**—Accepts the configuration exchange request set.
- **send**—Enables sending of the configuration exchange set.

Note The request and set options are enabled by default.

Step 9 **end**

Example:

```
Device(config-ikev2-profile)# end
```

Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Configuring the IKEv2 Name Mangler

Perform this task to specify the IKEv2 name mangler, which is used to derive a name for authorization requests and obtain AAA preshared keys. The name is derived from specified portions of different forms of remote IKE identities or the EAP identity. The name mangler specified here is referred to in the IKEv2 profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 name-mangler** *mangler-name*
4. **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**}
5. **eap** {**all** | **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} | **prefix** | **suffix** {**delimiter** {**.** | **@** | \}}}
6. **email** {**all** | **domain** | **username**}
7. **fqdn** {**all** | **domain** | **hostname**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 name-mangler** *mangler-name* **Example:** `Device(config)# crypto ikev2 name-mangler mangler1`	Defines a name mangler and enters IKEv2 name mangler configuration mode.
Step 4	**dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} **Example:** `Device(config-ikev2-name-mangler)# dn state`	Derives the name from any of the following fields in the remote identity of type DN (distinguished name). • **common-name** • **country** • **domain** • **locality** • **organization** • **organization-unit** • **state**

Step 5	**eap** {**all** | **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} | **prefix** | **suffix** {**delimiter** {**.** | **@** | \}}}
	Example:
	`Device(config-ikev2-name-mangler)# eap prefix delimiter @`
		Derives the name from the remote identity of type EAP (Extensible Authentication Protocol).
		• **all**—Derives the name from the entire EAP identity.
		• **dn**—Derives the name from any of the following fields in the remote EAP identity of type DN:
		  • **common-name**
		  • **country**
		  • **domain**
		  • **locality**
		  • **organization**
		  • **organization-unit**
		  • **state**
		• **prefix**—Derives the name from the prefix in the EAP identity.
		• **suffix**—Derives the name from the suffix in the EAP identity.

	Command or Action	Purpose
		• **delimiter** {**.** | **@** | \\}—Specifies the delimiter in the EAP identity that separates the prefix and the suffix.
Step 6	**email** {**all** | **domain** | **username**} **Example:** `Device(config-ikev2-name-mangler)# email username`	Derives the name from the remote identity of type e-mail. • **all**—Derives the name from the entire remote IKE identity of type e-mail. • **domain**—Derives the name from the domain part of the remote IKE identity. • **username**—Derives the name from the username part of the remote IKE identity.
Step 7	**fqdn** {**all** | **domain** | **hostname**} **Example:** `Device(config-ikev2-name-mangler)# fqdn domain`	Derives the name from the remote identity of type FQDN (Fully Qualified Domain Name). • **all**—Derives the name from the entire remote IKE identity of type FQDN. • **domain**—Derives the name from the domain part of the remote IKE identity. • **hostname**—Derives the name from the hostname part of the remote IKE identity.
Step 8	**end** **Example:** `Device(config-ikev2-name-mangler)# end`	Exits IKEv2 name mangler configuration mode and returns to privileged EXEC mode.

Configuring the IKEv2 Authorization Policy

Perform this task to configure the IKEv2 authorization policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 authorization policy** *policy-name*
4. **aaa attribute list** *list-name*
5. **backup-gateway** *string*
6. **banner** *banner-text*
7. **configuration url** *url*
8. **configuration version** *version*
9. **def-domain** *domain-name*
10. **dhcp** {**giaddr** *ip-address* | **server** {*ip-address* | *hostname*} | **timeout** *seconds*}
11. [**ipv6**] **dns** *primary-server* [*secondary-server*]
12. **include-local-lan**

13. **ipsec flow-limit** *number*
14. **netmask** *mask*
15. **pfs**
16. [**ipv6**] **pool** *name*
17. **route set** {**interface** *interface* | **access-list** {*access-list-name* | *access-list-number* | **ipv6** *access-list-name*}}
18. **route accept any** [**tag** *value*] [**distance** *value*]
19. **route redistribute** *protocol* [**route-map** *map-name*]
20. **route set remote** {**ipv4** *ip-address mask* | **ipv6** *ip-address/mask*}
21. **smartcard-removal-disconnect**
22. **split-dns** *string*
23. **session-lifetime** *seconds*
24. **route set access-list** {*acl-number* | [**ipv6**] *acl-name*}
25. **wins** *primary-server* [*secondary-server*]
26. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 authorization policy** *policy-name* **Example:** `Device(config)# crypto ikev2 authorization policy policy1`	Specifies the IKEv2 authorization policy and enters IKEv2 authorization policy configuration mode.
Step 4	**aaa attribute list** *list-name* **Example:** `Device(config-ikev2-author-policy)# aaa attribute list list1`	Specifies an AAA attribute list. **Note** The AAA attribute list referred to in this command should be defined in global configuration mode.
Step 5	**backup-gateway** *string* **Example:** `Device(config-ikev2-author-policy)# backup-gateway gateway1`	Allows you to specify up to ten backup server names. This parameter is pushed to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies the backup servers that the client can use.
Step 6	**banner** *banner-text* **Example:** `Device(config-ikev2-author-policy)# banner This is IKEv2`	Specifies the banner. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute.

	Command or Action	Purpose
Step 7	**configuration url** *url* **Example:** `Device(config-ikev2-author-policy)# configuration url http://www.cisco.com`	Specifies the configuration URL. This parameter is sent to the client via the nonstandard Cisco FlexVPN configuration attribute. The client can use this URL to download the configuration.
Step 8	**configuration version** *version* **Example:** `Device(config-ikev2-author-policy)# configuration version 2.4`	Specifies the configuration version. This parameter is sent to the client via the nonstandard Cisco FlexVPN configuration attribute. This parameter is sent with the configuration URL to specify the version that the client can download.
Step 9	**def-domain** *domain-name* **Example:** `Device(config-ikev2-author-policy)# def-domain cisco`	Specifies the default domain. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies the default domain that the client can use.
Step 10	**dhcp** {**giaddr** *ip-address* | **server** {*ip-address* | *hostname*} | **timeout** *seconds*} **Example:** `Device(config-ikev2-author-policy)# dhcp giaddr 192.0.2.1`	Specifies the DHCP server to lease an IP address that is assigned to the remote access client. • **giaddr** *ip-address*—Specifies the gateway IP address (giaddr). • **server** {*ip-address* | *hostname*}—Specifies the IP address or hostname of the DHCP server. The hostname is resolved during configuration. • **timeout** *seconds*—Specifies the wait time in seconds for the response from the DHCP server. **Note** You can specify only one DHCP server. It is assumed that the DHCP server can be reached via the global routing table, and therefore, the DHCP packets are forwarded to the global routing table.
Step 11	[**ipv6**] **dns** *primary-server* [*secondary-server*] **Example:** `Device(config-ikev2-author-policy)# dns 198.51.100.1 198.51.100.100`	Specifies the IP addresses of primary and secondary Domain Name Service (DNS) servers that are sent to the client in the configuration reply. • **ipv6**—(Optional) Specifies an IPv6 address for the DNS server. To specify an IPv4 address, execute the command without this keyword. • *primary-server*—IP address of the primary DNS server. • *secondary-server*—(Optional) IP address of the secondary DNS server.
Step 12	**include-local-lan** **Example:**	Includes local LAN. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute.

	Command or Action	Purpose
	`Device(config-ikev2-author-policy)# include-local-lan`	
Step 13	**ipsec flow-limit** *number* **Example:** `Device(config-ikev2-author-policy)# ipsec flow-limit 12500`	Specifies the maximum number of IPsec SAs that an IKev2 dVTI session on the IKev2 responder can have. The range is from 0 to 50000. By default, the command is disabled, and there is no limit on the number of IPsec flows per dVTI session. A value of 0 will not allow any IPsec SAs.
Step 14	**netmask** *mask* **Example:** `Device(config-ikev2-author-policy)# netmask 255.255.255.0`	Specifies the netmask of the subnet from which the IP address is assigned to the client. • *mask*—Subnet mask address.
Step 15	**pfs** **Example:** `Device(config-ikev2-author-policy)# pfs`	Enables Password Forward Secrecy (PFS). This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies whether the client should use PFS.
Step 16	[**ipv6**] **pool** *name* **Example:** `Device(config-ikev2-author-policy)# pool abc`	Defines a local IP address pool for assigning IP addresses to the remote access client. • **ipv6**—(Optional) Specifies an IPv6 address pool. To specify an IPv4 address, execute the command without this keyword.. • *name*—Name of the local IP address pool. **Note** The local IP address pool must already be defined using the **ip local pool** command.
Step 17	**route set** {**interface** *interface* | **access-list** {*access-list-name* | *access-list-number* | **ipv6** *access-list-name*}} **Example:** `Device(config-ikev2-author-policy)# route set interface`	Specifies the route set parameters to the peer via configuration mode and allows running routing protocols such as Border Gateway Protocol (BGP) over VPN. • **interface**—Specifies the route interface. • **access-list**—Specifies the route access list. • *access-list-name*—Access list name. • *access-list-number*—Standard access list number. • **ipv6**—Specifies an IPv6 access list.
Step 18	**route accept any** [**tag** *value*] [**distance** *value*] **Example:** `Device(config-ikev2-author-policy)# route accept any tag 10`	Filters the routes received from the peer and specify the tag and metric values to install these routes. • **any**—Accepts all routes received from the peer. • **tag** *value*—(Optional) Specifies the tag ID for the static routes added by IKEv2. The range is from 1 to 497777.

	Command or Action	Purpose
		• **distance** *value*—(Optional) Specifies the distance for the static routes added by IKEv2. The range is from 1 to 255.
Step 19	**route redistribute** *protocol* [**route-map** *map-name*] **Example:** `Device(config-ikev2-author-policy)# route redistribute connected`	Filters the routes received from the peer and specify the tag and metric values to install these routes. • *protocol*—Source protocol from which routes are redistributed. It can be one of the following keywords: connected or static. • **route-map** *map-name*—(Optional) Route map that should be filtered to import routes from one source routing protocol to another routing protocol. If a map name is not specified, all routes are redistributed.
Step 20	**route set remote** {**ipv4** *ip-address mask* | **ipv6** *ip-address/mask*} **Example:** `Device(config-ikev2-author-policy)# route set remote ipv6 2001:DB8::1/32`	Configures IP addresses of inside networks.
Step 21	**smartcard-removal-disconnect** **Example:** `Device(config-ikev2-author-policy)# smartcard-removal-disconnect`	Enables smartcard removal disconnect. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies that the client should terminate the session when the smart card is removed.
Step 22	**split-dns** *string* **Example:** `Device(config-ikev2-author-policy)# split-dns abc1`	Allows you to specify up to ten split domain names. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies the domain names that the client should use for private networks.
Step 23	**session-lifetime** *seconds* **Example:** `Device(config-ikev2-author-policy)# session-lifetime 1000`	Specifies the IKEv2 session lifetime. • **seconds** *seconds*—The range is from 120 to 25920000, which converts to two minutes to 300 days.
Step 24	**route set access-list** {*acl-number* | [**ipv6**] *acl-name*} **Example:** `Device(config-ikev2-client-config-group)# route set access-list 110`	Specifies the subnets that are pushed to the remote peer via configuration mode. • *acl-number*—Access list number (ACL). The ACL number can only be specified for an IPv4 ACL. • **ipv6**—(Optional) Specifies an IPv6 access control list (ACL). To specify an IPv4 attribute, execute the command without this keyword. • *acl-name*—Access list name.

	Command or Action	Purpose
		Note You can only specify standard, simple access lists for IPv4 addresses.
Step 25	**wins** *primary-server* [*secondary-server*] **Example:** `Device(config-ikev2-author-policy)# wins 203.0.113.1 203.0.113.115`	Specifies the internal Windows Internet Naming Service (WINS) server addresses that are sent to the client in the configuration reply. • *primary-server*—IP address of the primary WINS server. • *secondary-server*—(Optional) IP address of the secondary WINS server.
Step 26	**end** **Example:** `Device(config-ikev2-author-policy)# end`	Exits IKEv2 authorization policy configuration mode and returns to privileged EXEC mode.

Configuration Examples for the FlexVPN Server

Example: Configuring the FlexVPN Server

Example: Configuring the FlexVPN Server to Authenticate Peers Using EAP

This example shows how to configure the FlexVPN server to authenticate peers using EAP.

```
aaa new-model
!
aaa group server radius eap-server
 server 192.168.2.1
!
aaa authentication login eap-list group eap-server
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto ikev2 profile ikev2-profile1
 match identity remote address 0.0.0.0
 authentication local rsa-sig
 authentication remote eap query-identity
 pki trustpoint trustpoint1
 aaa authentication eap eap-list
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
```

```
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.1 key key1
!
```

Example: Configuring the FlexVPN Server for Group Authorization (External AAA)

The following example shows how to configure the FlexVPN server for group authentication through an external AAA, which would be the RADIUS or TACACS server.

```
aaa new-model
!
aaa group server radius cisco-acs
 server 192.168.2.2
!
aaa authorization network group-author-list group cisco-acs
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 name-mangler group-author-mangler
 dn domain
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list group-author-list name-mangler group-author-mangler
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.2 key key2
!
```

Example: Configuring the FlexVPN Server for Group Authorization (Local AAA)

The following example shows how to configure the FlexVPN server for group authorization through the local AAA using the IKEv2 authorization policy. The authorization policy specifies standard IPv4 and IPv6 attributes,

and Cisco Unity, and FlexVPN attributes to be sent to the client through configuration mode. The authorization policy also specifies per user attributes through **aaa attribute list** command for local use.

```
aaa new-model
!
aaa authorization network local-group-author-list local
!
!
aaa attribute list attr-list1
 attribute type interface-config "ip mtu 1100"
 attribute type interface-config "tunnel key 10"
!

crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 authorization policy author-policy1
 pool pool1
 dhcp server 192.168.4.1
 dhcp timeout 10
 dhcp giaddr 192.168.1.1
 dns 10.1.1.1 10.1.1.2
 route set access-list acl1
 wins 192.168.1.2 192.168.1.3
 netmask 255.0.0.0
 banner ^C flexvpn server ^C
 configuration url http://www.abc.com
 configuration version 10
 def-domain abc.com
 split-dns dns1
 split-dns dns2
 split-dns dns3
 backup-gateway gw1
 backup-gateway gw2
 backup-gateway gw3
 smartcard-removal-disconnect
 include-local-lan
 pfs
 aaa attribute list attr-list1
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list local-group-author-list author-policy1
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
```

```
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
ip local pool pool11 192.168.2.10 192.168.2.100
!
ip access-list extended acl-1
 permit ip 192.168.3.10 192.168.4.100 any
 permit ip 192.168.10.1 192.168.10.100 any
!
```

Example: Configuring the FlexVPN Server for User Authorization

The following example shows how to configure the FlexVPN server for user authentication.

```
aaa new-model
!
aaa group server radius cisco-acs
 server 192.168.2.2
!
aaa authorization network user-author-list group cisco-acs
!
crypto pki trustpoint trustpoint1
 enrollment url http:// 192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 name-mangler user-author-mangler
 dn common-name
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization user cert list user-author-list name-mangler user-author-mangler
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.2 key key2
!
```

Example: Configuring the FlexVPN Server for IPv6 Session with IPv6 Configuration Attributes

The following example shows how to configure the FlexVPN server for an IPv6 dynamic Virtual Tunnel Interfaces (dVTI) session. The example uses local AAA group authorization using the IKEv2 authorization policy. The IPv6 configuration attributes are configured under the IKEv2 authorization policy.

```
aaa new-model
!
aaa authorization network local-group-author-list local
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 authorization policy author-policy1
 ipv6 pool v6-pool
 ipv6 dns 2001:DB8:1::11 2001:DB8:1::12
 ipv6 subnet-acl v6-acl
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list local-group-author-list author-policy1
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ipv6 address 2001:DB8:1::1/32
!
interface Virtual-Template1 type tunnel
 ipv6 unnumbered Ethernet0/0
 tunnel mode ipsec ipv6
 tunnel protection ipsec profile ipsec-profile1
!
ipv6 local pool v6-pool 2001:DB8:1::10/32 48
!
ipv6 access-list v6-acl
 permit ipv6 host 2001:DB8:1::20 any
 permit ipv6 host 2001:DB8:1::30 any
!
```

Example: Configuring AnyConnect Profile Download

The following example shows how to configure the FlexVPN AnyConnect Profile Download feature:

Note You do not modify the Local Policy files on the Anyconnect Client machine. After the configuration of Anyconnect Profile Download feature on IKEv2, the required XML profiles get automatically downloaded on the client device.

Note You should disable either the HTTPS server (ip http secure-server) or SSL policy (crypto ssl policy) for the profile download feature, otherwise, if both these features are enabled at the same time and the device receives an incoming SSL VPN connection, the device may crash.

```
no ip http secure-server
crypto ssl policy ssl-policy
  pki trustpoint CA1 sign
  ip address local 10.0.0.1 port 443
  no shutdown
crypto ssl profile ssl_prof
    match policy ssl-policy
crypto vpn anyconnect profile ANY-PROF bootflash:profile.xml
crypto ikev2 profile ikev2_profile
    anyconnect profile ANY-PROF
```

Additional References for Configuring the FlexVPN Server

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Cisco AnyConnect Secure Mobility Client	https://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client/tsd-products-support-series-home.html
IPsec configuration	*Configuring Security for VPNs with IPsec*
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring the FlexVPN Server

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 276: Feature Information for Configuring the FlexVPN Server

Feature Name	Releases	Feature Information
IKEv2 headend support for remote access clients	Cisco IOS XE Release 3.5S	This feature provides IKEv2 support for Anyconnect 3.0, FlexVPN hardware client, and multi SA support for VTI. The following commands were introduced or modified: **aaa attribute list, backup-gateway, banner, config-mode set, configuration url, configuration version, def-domain, dhcp, dns, include-local-lan, max flow limit, pfs, pool, route accept, route set interface, smartcard-removal-disconnect, split-dns, subnet-acl.**

CHAPTER 208

Configuring the FlexVPN Client

This module describes the FlexVPN client features and the Internet Key Exchange Version 2 (IKEv2) commands required to configure the FlexVPN client.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Restrictions for the FlexVPN Client

EAP as the Local Authentication Method

- Extensible Authentication Protocol (EAP) as the local authentication method, is supported only on the IKEv2 initiator, and as the remote authentication, is supported only on the IKEv2 responder.
- If EAP is specified as the local authentication method, the remote authentication method must be certificate based.
- If the **authentication remote eap query-identity** command is not configured on the FlexVPN server, the client cannot have an IPv4 or IPv6 address as the local identity because these IP addresses cannot be used as the username for the EAP authentication method.

Dual-Stack Tunnel Interface and VRF-Aware IPsec

When configuring a dual-stack tunnel interface in a VPN routing and forwarding (VRF)-aware IPsec scenario, you cannot use the **ip vrf forwarding** command to configure an Inside VPN routing and forwarding (IVRF)

instance because this is not a valid configuration. Use the **vrf forwarding** *vrf-name* command to define the IVRF of the tunnel interface, where the *vrf-name* argument is defined using the **vrf definition** command with IPv4 and IPv6 address families inside the definition.

SSO Restrictions

- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.

Information About the FlexVPN Client

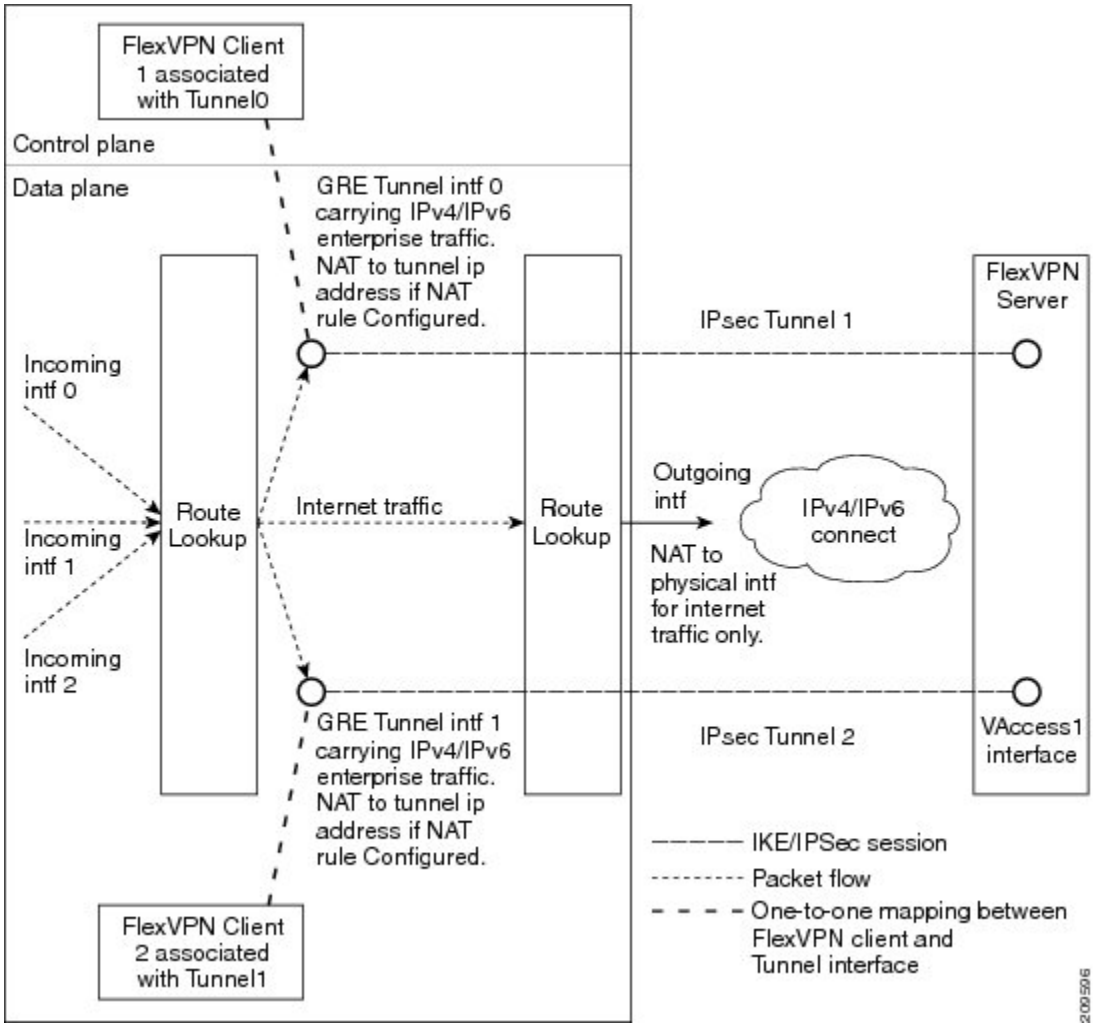
IKEv2 FlexVPN Client

The IKEv2 FlexVPN Client feature establishes a secure IPsec VPN tunnel between a FlexVPN client and a FlexVPN server. The IKEv2 FlexVPN Client feature provides the following benefits:

- Unified tunnel infrastructure
- IPv4/IPv6 proxy support over IPv4/IPv6 transport
- Backward compatibility with some features supported by EasyVPN
- Flexibility for running dynamic routing protocols

Each FlexVPN client is associated with a unique tunnel interface, which implies that the IPsec security association (SA) retrieved by the specific FlexVPN client is bound to the tunnel interface. The figure below shows the association between the FlexVPN client and the tunnel interface.

Figure 107: Association of the FlexVPN Client and the Tunnel Interface



The sequence of operation is as follows:

- Routing—The FlexVPN server pushes the network list as part of the mode configuration response. The client adds routes on the tunnel interface to these networks. As part of the configuration mode set, the client sends the routes to its network. The IP address is configured on the tunnel interface so that the server can add routes to the client-side network.
- NAT—Network Address Translation (NAT) rules must be configured explicitly using route maps. If the rules match, the hosts behind the FlexVPN client are translated to the tunnel IP address. This IP address can be obtained as one of the attributes pushed during mode configuration by the FlexVPN server.
- Encapsulation and encryption—Generic routing encapsulation (GRE) and IPsec encapsulation modes are supported. GRE supports both IPv4 and IPv6 traffic. The traffic that reaches the tunnel interface is encapsulated by the GRE header, followed by IPsec protection. The encrypted traffic is then routed to the outgoing interface.

The features supported by the FlexVPN client are described in the following sections:

Tunnel Activation

The FlexVPN client can be connected automatically or manually through user intervention. The FlexVPN client connects automatically to the tunnel when the FlexVPN configuration is complete. If the tunnel times out or fails, the tunnel automatically reconnects and retries the connection indefinitely. To configure an automatic tunnel connection, use the **connect** command with the **auto** keyword in the IKEv2 FlexVPN profile.

In a manual connection, the FlexVPN client waits for user intervention to execute a command before establishing a connection. When the client times out or fails to connect, subsequent connections require user intervention. To configure a manual connection, use the **crypto ikev2 client flexvpn connect** command with the *flexvpn-name* argument in privileged EXEC mode. To terminate the connection, use the **clear crypto ikev2 client flexvpn connect** command with the *flexvpn-name* argument.

Tracking-Based Tunnel Activation

The Tracking-Based Tunnel Activation feature is mainly used in backup scenarios. The FlexVPN client registers with the tracking system to obtain notifications for change in the state of an object. This notification prompts the client to perform an appropriate action for tunnel activation. The **track** keyword in the **connect** command informs the tracking process that the client is interested in tracking an object, which is identified by an object number. The tracking process, in turn, informs the client when the state of the objects changes.

If the **track** keyword in the **connect** command is set to activate the tunnel when the object goes up, the client triggers the connection upon receiving the notification that the object is in the UP state. If the **track** keyword in the **connect** command is set to activate the tunnel when the object goes down, the client triggers the connection upon receiving the notification that the object is in the DOWN state.

Backup Features

A FlexVPN client can connect to various peers or servers in a predetermined order. The list of peers is called the gateway list or backup gateway list and is built using the following lists:

- Static backup gateway list or static list
- Downloaded backup gateway list or downloaded list

The static backup gateway list is configured in the FlexVPN profile by providing a list of peers with a sequence number. The downloaded backup gateway list is downloaded dynamically and is obtained during the mode configuration response. The downloaded list complements the static gateway list to build the backup gateway list. The downloaded list is inserted after the peer from which the list is downloaded.

If an existing connection with a peer from the gateway list goes down, the client tries to establish a connection with the next peer in the gateway list. If a downloaded list is available and connection with a static peer fails, the client tries to connect, in sequence, with the peers from the downloaded list. If the client fails to establish a connection with all the peers in the downloaded list, the client tries to connect to the next peer in the static list, and the downloaded list is deleted.

Backup Gateways

Use the **peer** command to add a peer to the backup gateway list. To remove the backup gateway list, use the **no peer** command.

Peers are ordered by preference; the lower the sequence number, the higher the preference.

If a connection is established with a new peer and the peer is not a part of the downloaded list, the peer adds the downloaded list to the backup gateway list, and the existing backup gateway list is replaced with the new list.

You can configure a static peer and attach it to a track object. A peer is a "possible peer" if the track object of the peer is in the UP state.

Note Peers that are not attached to a track object, including peers in the downloaded list, are classified as "possible peers" because these peers are always in the UP state.

The peer selection process works as follows: when a connection is established, the gateway list is looked up and the first possible peer is selected. A peer is selected according to the following rule: a static peer can be associated with the track object with a desired status (UP or DOWN). If the status of the track object matches the configured status, the peer is said to be a "possible peer."

Note If the peer is identified by either a Domain Name Service (DNS) name or a fully qualified domain name (FQDN), the name is resolved dynamically.

The peer selection process is followed by the selection of a new peer or when the existing criteria fail, which happens in the following scenarios:

- The active peer stops responding to liveness checks.
- The DNS resolution of the peer name fails.
- The IKE negotiation with the peer fails.
- The peer is no longer a "possible peer" (its corresponding track object goes DOWN).

Note When you configure multiple FlexVPN peers on a FlexVPN client and when you clear the IKEv2 SA on the primary peer, the clearance will trigger a new peer selection on the client.

Reactivate Primary Peer

The Reactivate Primary Peer feature ensures that the highest-priority peer is always connected. If the track object of the highest-priority peer matches the object status, the existing connection with the lower-priority peer is disconnected, and the connection to the higher-priority peer is established. Use the **peer reactivate** command to enable this feature.

Note A track object must be associated with statically configured peers.

Dial Backup (Primary or Backup Tunnels)

The FlexVPN client registers with the tracking system to get notifications about the change in the state of the object. The **connect track** command is used to inform the tracking process that the client is interested in tracking an object, which is identified by the object number. The tracking process, in turn, informs the client when the state of this objects changes. This notification prompts the client to take further action to bring up or bring down the primary or backup connections when the state of the tracked object is UP or DOWN.

The Dial Backup feature can be configured as follows:

- When both primary and backup tunnels are FlexVPN tunnels,
  - Any one tunnel is active at a time.
  - Both client profiles are configured using the **connect track** command, referencing the same track object.
  - If the primary tunnel tracks the status when the object is UP, the secondary tunnel tracks the status of the object when the object is DOWN.
- When one tunnel is the FlexVPN tunnel,
  - The remaining tunnels can be on any secured connection.
  - The primary connection is not FlexVPN, and the backup connection is FlexVPN.
  - The client profile is configured using the **connect track** command with an object, which traces the ability to reach the primary peer through the primary outgoing interface.

Backup Group

The Backup Group feature allows the FlexVPN client to omit a peer when a FlexVPN client that belongs to a group has established a session with the same peer. When a FlexVPN client belonging to a group initiates a connection with a peer, the FlexVPN client validates if another FlexVPN client in the same group has established a session with the same peer. If a connection exists, the FlexVPN client omits this peer and validates the next peer in the sequence. Use the **backup group** command with the *group-number* argument to configure the backup group.

Dual FlexVPN Support

The Dual FlexVPN Support feature provides the ability to configure two FlexVPN tunnels that share the same inside and outside interfaces. The two FlexVPN tunnels use route injections to direct appropriate traffic through the corresponding tunnel interface. When the tunnel is up, the tunnel “learns” the network list from the server. If the server forwards a network list, FlexVPN installs specific routes to the destination networks in its routing table, directing the traffic to these networks out of the tunnel interface.

Note Only one FlexVPN connection can be established with a default route through the tunnel interface.

Split DNS Support

The Split DNS functionality enables the FlexVPN client to act as a Domain Name System (DNS) proxy. During FlexVPN negotiations, the DNS list is downloaded during mode configuration. This list is configured as a DNS view list on the inside interfaces associated with the FlexVPN profile. The view list is used to match requests based on the domain names with the DNS query and then forward the match requests to the DNS server. Other DNS queries are used to match the default view (global DNS configuration) and are forwarded to the ISP DNS.

If no inside interfaces are mentioned in the FlexVPN client profile, the DNS view is applied to all interfaces except the tunnel interface and the tunnel source interfaces of all configured profiles. When the DNS query request reaches the inside interface, the matching DNS view is obtained, and the request is forwarded to the DNS IP address.

NAT

The Network Address Translation (NAT) feature in FlexVPN enables traffic to be translated to an IP address based on the interface to which the traffic is routed. If a packet is received on one interface that is configured with the **ip nat inside** command and is being sent out another interface that is configured with the **ip nat outside** command, the packet is translated to the IP address configured on the second interface.

Network List from the Server

Routes for enterprise traffic are dynamically installed by a client through the tunnel interface. The traffic takes the default route via the outgoing physical interface. The enterprise traffic is translated to the tunnel IP address, and the Internet traffic is translated to the external outgoing interface IP address.

Default Route List from the Server

A default route must be configured on the device with the higher sequence number via the tunnel interface. The tunnel interface is configured with the **ip nat outside** command, and the IP address of the tunnel interface is assigned by the IP address sent by the client. The enterprise traffic from inside interfaces is translated to the sent address. NAT is achieved by configuring NAT rules with the help of route maps. The route maps define rules based on the outgoing interface, by which the globally configured NAT rules are applied based on routing.

IPv4 traffic going out the tunnel interface is translated to the sent IPv4 address.

Note If NAT is not required, NAT rules associated with the tunnel interface must not be configured.

How the FlexVPN Client learns about the Network List

The FlexVPN client learns about the list of networks behind a peer in one of the following ways:

- Mode configuration push—The FlexVPN server sends the list of network attributes as a configuration mode parameter to the client. The FlexVPN client installs the routes to these networks through the tunnel interface that has the highest metric. The client also communicates its networks to the server in the mode configuration set or acknowledgment (SET/ACK) exchange so that the server can add those routes via the virtual access interface.
- Running routing protocols—The FlexVPN client and server run routing protocols over the tunnel interface to establish network routes, which allows the client and the server the flexibility to add or remove networks without disconnecting the existing session. The tunnel addresses are communicated during mode configuration to establish routes with peers.

WINS NBNS and DOMAIN Name

The FlexVPN server pushes the domain name, Windows Internet Naming Service (WINS), or NetBios Name Server (NBNS) attributes during mode configuration. These attributes are dynamically updated to the DHCP server that runs on the FlexVPN client.

Event Tracing

The Event Tracing feature is used for debugging purposes. Events posted to the FlexVPN client are logged, and the information is used for debugging. Event tracing is a combination of a fast mechanism that logs a few

bytes of trace information in a buffer area and a display mechanism that extracts and decodes the debug data. The FlexVPN client maintains its buffer and can be enabled during normal operation.

Extensible Authentication Protocol as a Local Authentication Method

The FlexVPN client supports EAP as a local authentication method. Supported EAP authentication methods are Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAPv2), message digest algorithm 5 (MD5), and Generic Token Card (GTC). The EAP authentication process is as follows:

- Use the **authentication local eap** command in IKEv2 profile configuration mode to authenticate the FlexVPN client by using EAP.
- After the FlexVPN client receives the IKE_AUTH response from the peer, enter the **crypto eap credentials** command.
- If the EAP-Identity Request is received in the IKE_AUTH response, the EAP username and password must be specified.
- If an EAP-Identity Request is not received in the IKE_AUTH response, only the password is specified because the local IKEv2 identity is used as the username.

Note EAP as the local authentication method must be used with the FlexVPN client, but EAP can also be used on the IKEv2 initiator. If the EAP server initially proposes an unsupported authentication method, the FlexVPN EAP initiator responds with an EAP Negative Acknowledgment (NAK) packet, requesting EAP-MSCHAPv2, EAP-MD5, or EAP-GTC as the desired authentication method. The FlexVPN EAP responder selects one of the authentication methods.

How to Configure the FlexVPN Client

Configuring the IKEv2 VPN Client Profile

This task describes the IKEv2 commands required for configuring the FlexVPN client and the basic IKEv2 commands. Refer to the “Configuring Basic Internet Key Exchange Version 2 CLI Constructs” task in the *Configuring Internet Key Exchange Version 2 (IKEv2)* module for information about configuring the basic IKEv2 profile.

Note When you enter a typo in authorization list under ikev2 profile, it automatically goes back to the default list.

Refer to the “How to Configure the FlexVPN Client” section for information about configuring an IKEv2 profile for the FlexVPN server.

Configuring the Tunnel Interface

Perform this task to configure the tunnel interface that is referred to by the FlexVPN client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** {*ipv4-address* | **negotiated**}
5. **tunnel mode gre ip**
6. **tunnel mode ipsec ipv4**
7. **tunnel source** {*ip-address* | *interface* | **dynamic**}
8. **tunnel destination dynamic**
9. **tunnel protection ipsec-profile** *profile-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 1`	Creates a tunnel interface and enters interface configuration mode.
Step 4	**ip address** {*ipv4-address*	**negotiated**} **Example:** `Device(config-if)# ip address negotiated`	(Optional) Assigns an IPv4 address to the tunnel interface.
Step 5	**tunnel mode gre ip** **Example:** `Device(config-if)# tunnel mode gre ip`	(Optional) Enables generic route encapsulation (GRE) mode for the tunnel interface.
Step 6	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	(Optional) Enables IPsec encapsulation.
Step 7	**tunnel source** {*ip-address*	*interface*	**dynamic**} **Example:** `Device(config-if)# tunnel source 10.0.0.1`	Specifies the source for the tunnel interface.
Step 8	**tunnel destination dynamic** **Example:**	Specifies the destination for the tunnel interface.

	Command or Action	Purpose
	`Device(config-if)# tunnel destination dynamic`	
Step 9	**tunnel protection ipsec-profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec-profile ipsecprofile1`	Associates a tunnel interface with an IPsec profile.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the FlexVPN Client

Use the **monitor event-trace flexvpn** command to enable event tracing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 client flexvpn** *client-name*
4. **peer** *sequence* {*ipv4-address* | *ipv6-address* | **fqdn** *fqdn-name* [**dynamic** | **ipv6**]} [**track** *track-number* [**up** | **down**]]
5. **connect** {**manual** | **auto** | **track** *track-number* [**up** | **down**]}
6. **client inside** *interface-type interface-number*
7. **client connect tunnel** *interface-number*
8. **source** *sequence-number interface-type interface-number* **track** *track-number*
9. **peer reactivate**
10. **backup group** {*group-number* | **default**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 client flexvpn** *client-name* **Example:** `Device(config)# crypto ikev2 client flexvpn client1`	Defines an IKEv2 FlexVPN client profile and enters IKEv2 FlexVPN client profile configuration mode.

	Command or Action	Purpose
Step 4	**peer** *sequence* {*ipv4-address* | *ipv6-address* | **fqdn** *fqdn-name* [**dynamic** | **ipv6**]} [**track** *track-number* [**up** | **down**]] **Example:** `Device(config-ikev2-flexvpn)# peer 1 10.0.0.1`	Defines a static peer using an IP address or hostname.
Step 5	**connect** {**manual** | **auto** | **track** *track-number* [**up** | **down**]} **Example:** `Device(config-ikev2-flexvpn)# connect track 10 up`	Connects the FlexVPN tunnel. **Note** Any change to this command terminates the active session.
Step 6	**client inside** *interface-type interface-number* **Example:** `Device(config-ikev2-flexvpn)# client inside GigabitEthernet 0/1`	(Optional) Specifies the inside interface. • You can specify more than one inside interface in a FlexVPN client profile. The inside interfaces can be shared across FlexVPN client profiles. **Note** Any change to this command terminates the active session.
Step 7	**client connect tunnel** *interface-number* **Example:** `Device(config-ikev2-flexvpn)# client connect tunnel 1`	Assigns the tunnel interface created in the "Configuring the Tunnel Interface" task to the FlexVPN client. • You can configure only one tunnel interface for a FlexVPN client profile. **Note** Any change to this command terminates the active session.
Step 8	**source** *sequence-number interface-type interface-number* **track** *track-number* **Example:** `Device(config-ikev2-flexvpn)# source 1 GigabitEthernet 0/1 track 11`	Adds sequence numbers to the tunnel source address. • The tunnel source address has the lowest sequence number for which the track object number is in UP state. **Note** Any change to this command terminates the active session.
Step 9	**peer reactivate** **Example:** `Device(config-ikev2-flexvpn)# peer reactivate`	Enables the reactivate primary peer feature.
Step 10	**backup group** {*group-number* | **default**} **Example:** `Device(config-ikev2-flexvpn)# backup group default`	Assigns the client to a backup group. • By default, all clients belong to backup group 0. **Note** Any change to this command terminates the active session.

	Command or Action	Purpose
Step 11	**end** **Example:** `Device(config-ikev2-flexvpn)# end`	Exits IKEv2 FlexVPN client profile configuration mode and returns to privileged EXEC mode.

Configuring EAP as the Local Authentication Method

Perform this task to configure Extensible Authentication Protocol (EAP) as the local authentication method on the FlexVPN client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **authentication local eap**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**authentication local eap** **Example:** `Device(config-ikev2-profile)# authentication local eap`	Specifies EAP as the local authentication method. **Note** This command is supported only on the IKEv2 initiator.
Step 5	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Configuration Examples for the FlexVPN Client

Example: Configuring the IKEv2 FlexVPN Client Profile

The following example shows how to configure the IKEv2 FlexVPN client profile:

```
crypto ikev2 client flexvpn flex
  peer 1 10.0.0.1
  connect manual
  client connect Tunnel0
!
crypto ikev2 authorization policy flex
 subnet-acl 199
 route set interface
 route accept any
!
crypto ikev2 keyring key
 peer dvti
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco
 !
crypto ikev2 profile prof
 match identity remote address 10.0.0.1 255.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring key
 aaa authorization group psk list local-group-author-list flex
 config-mode set
!
crypto ipsec transform-set trans esp-aes
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set pfs group2
 set ikev2-profile prof
!
interface Tunnel0
 ip address negotiated
 tunnel source Ethernet0/0
 tunnel destination dynamic
 tunnel mode ipsec ipv4
 tunnel protection ipsec-profile ipsecprof
!
interface Ethernet0/0
 ip address 172.16.0.1 255.240.0.0
 ip virtual-reassembly in
!
 ip route 0.0.0.0 0.0.0.0 2.2.2.2
access-list 199 permit ip 10.20.20.20 0.0.0.255 any
access-list 199 permit ip 10.30.30.30 0.0.0.255 any
```

Example: Configuring EAP as a Local Authentication Method

The following example shows how to configure EAP as a local authentication method:

```
crypto ikev2 profile profile1
 authentication remote rsa-sig
 authentication local eap
```

When the session is brought up, a prompt appears to enter the EAP credentials, as follows:

```
Enter the command "crypto eap credentials profile1"
Device# crypto eap credentials profile1

Enter the Username for profile profile1: cisco
Enter the password for username cisco
```

Additional References for Configuring the FlexVPN Client

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	*Configuring Security for VPNs with IPsec*
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring the FlexVPN Client

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 277: Feature Information for Configuring FlexVPN Client

Feature Name	Releases	Feature Information
IKEv2 Remote Access Hardware Client		The IKEv2 Remote Access Hardware Client feature provides support for remote access connectivity and the extensions necessary to support diverse solutions such as mobility, NAT traversal, reliability, and enhanced denial of service (DoS) attack resilience. The following commands were introduced or modified: **backup group, client connect tunnel, client inside, connect, crypto ikev2 client flexvpn, interface, ip address, peer, peer reactivate, source tunnel destination, tunnel mode, tunnel protection, tunnel source.**
IPv6 Remote Access for IPsec VPN		The IPv6 Remote Access for IPsec VPN feature provides IPv6 support and support for EAP as the local authentication method for the IKEv2 FlexVPN client. The following commands were modified: **authentication (IKEv2 profile), peer.**

CHAPTER 209

Configuring FlexVPN Spoke to Spoke

Last Published Date: March 28, 2014

The FlexVPN Spoke to Spoke feature enables a FlexVPN client to establish a direct crypto tunnel with another FlexVPN client leveraging virtual tunnel interfaces (VTI), Internet Key Exchange Version 2 (IKEv2), and Next Hop Resolution Protocol (NHRP) to build spoke-to-spoke connections.

Prerequisites for FlexVPN Spoke to Spoke

IKEv2, the FlexVPN server, and the FlexVPN spoke must be configured.

Information About FlexVPN Spoke to Spoke

FlexVPN and NHRP

FlexVPN is Cisco's implementation of the IKEv2 standard featuring a unified paradigm and CLI that combines site to site, remote access, hub and spoke topologies and partial meshes (spoke to spoke direct). FlexVPN offers a simple but modular framework that extensively uses the tunnel interface paradigm while remaining compatible with legacy VPN implementations using the crypto maps.

The FlexVPN server provides the server side functionality of FlexVPN. The FlexVPN client establishes a secure IPsec VPN tunnel between a FlexVPN client and another FlexVPN server.

NHRP is an Address Resolution Protocol (ARP)-like protocol that alleviates nonbroadcast multiaccess (NBMA) network problems. With NHRP, NHRP entities attached to an NBMA network dynamically learn the NBMA address of the other entities that are part of that network, allowing these entities to directly communicate without requiring traffic to use an intermediate hop.

The FlexVPN Spoke to Spoke feature integrates NHRP and FlexVPN client (spoke) to establish a direct crypto channel with another client in an existing FlexVPN network. The connections are built using virtual tunnel interfaces (VTI), IKEv2 and NHRP, where NHRP is used for resolving the FlexVPN clients in the network.

The following is recommended in FlexVPN:

- Routing entries are not exchanged between spokes.
- Different profiles are used for the spokes and the **config-exchange** command is not configured for the spokes.

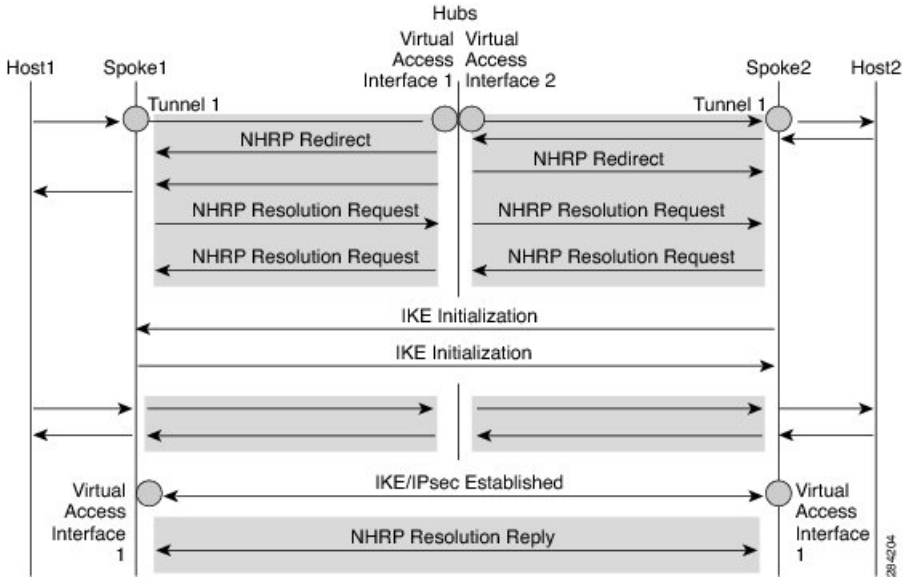
The FlexVPN IPv6 Direct Spoke to Spoke feature supports the use of IPv6 addresses for FlexVPN spokes. The support for IPv6 addresses provides support for IPv6 over IPv4, IPv4 over IPv6, and IPv6 over IPv6 transports.

Note Spoke to Spoke FlexVPN does not support dynamic AAA authorization.

NHRP Resolution Request and Reply in FlexVPN

The following diagram illustrates the NHRP resolution request and reply in FlexVPN.

Figure 108: NHRP Resolution Request and Reply



Due to bidirectional traffic, similar events occur in both directions at Spoke1, Spoke2, and hub. For clarity, events from Host1 to Host2 are discussed. Assume that there is a network N1 (192.168.1.0/24) behind Spoke1 and another network N2 (192.168.2.0/24) behind Spoke2. The network between the two spokes is matched through an access control list (ACL). This is because ACLs are applied on the IKEv2 policies on both spokes.

The network along with its prefix information from both the spokes is conveyed to the hub via IKEv2 information payload exchanges. This causes a route addition in the routing table by IKEv2 at the hub as follows:

- 192.168.1.0/24—Connected via virtual access interface1

- 192.168.2.0/24—Connected via virtual access interface2

The hub will push a summarized route via IKEv2 to both spokes, and the spokes will install the route in their routing table as follows:

- 192.168.0.0/16—next hop <tunnel address of the hub> - interface Tunnel 1

Note The routing protocol can also add the route to the routing table.

Assuming that traffic moves from N1 to N2, the traffic flow is as follows:

1. Host1 sends traffic destined to Host2. The traffic reaches the LAN interface of spoke1, looks up the route, hits the summarized route, and routes the packet to interface tunnel 1.
2. When the traffic reaches the hub's virtual access interface1, the traffic looks up the route table for a route entry for N2, either directly connected over virtual access interface 2 or via a point-to-point tunnel interface.
3. The traffic from Host1 to Host2 traverses the hub through virtual access interface1 and virtual access interface2. The hub determines that ingress and the egress interfaces (virtual access interface1 and virtual access interface2) belong to same NHRP network (network D configured on both the interfaces). The hub sends out an NHRP redirect message to spoke1 on virtual access interface1.
4. On receiving the redirect, Spoke1 initiates a resolution request for Host2 over the point-to-point tunnel interface (the same interface over which it received the redirect). The resolution request traverses the routed path (Spoke1-hub-spoke2). On receiving the resolution request, Spoke2 determines that it is the exit point and needs to respond to the resolution request.
5. Spoke2 receives the resolution request on the tunnel interface and retrieves the virtual template number from the tunnel interface. The virtual template number is used to create the virtual access interface to start a crypto channel and establishes IKEv2 and IPsec security associations (SAs). Once the crypto SAs between the two spokes are up, Spoke2 installs the necessary NHRP cache entries for Spoke1 and its network under the newly created virtual access interface and sends out the resolution reply over the virtual access interface.
6. After receiving the resolution request over the virtual access interface, Spoke1 installs the necessary cache entries for Spoke2 and its network. Spoke1 also deletes the temporary cache entry pointing to the hub to resolve the network under tunnel interface1.
7. NHRP adds shortcut routes as next-hop override (NHO) or H route. For more information on shortcut switching, refer to Shortcut Switching Enhancements for NHRP in DMVPN Networks.

How to Configure FlexVPN Spoke to Spoke

Configuring the Virtual Tunnel Interface on the FlexVPN Server

Before you begin

The FlexVPN server and client must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number* **type tunnel**
4. **ip unnumbered loopback** *number*
5. Do one of the following:
   - **ip nhrp network-id** *number*
   - **ipv6 nhrp network-id** *number*
6. **ip nhrp redirect** [**timeout** *seconds*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **type tunnel** **Example:** `Device(config)# interface virtual-template 1 type tunnel`	Creates a virtual template interface that can be configured and applied dynamically to create virtual access interfaces.
Step 4	**ip unnumbered loopback** *number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Assigns the IP address of an existing interface (usually a loopback interface) to the virtual tunnel interface.

Step 5	Do one of the following:
	• **ip nhrp network-id** *number*
	• **ipv6 nhrp network-id** *number*
	Example:
	`Device(config-if)# ip nhrp network-id 1`
	Example:
	`Device(config-if)# ipv6 nhrp network-id 1`
		Enables NHRP on the interface.

Step 6	**ip nhrp redirect** [**timeout** *seconds*] **Example:** `Device(config-if)# ip nhrp redirect`	Enables redirect traffic indication if traffic is forwarded with the NHRP network. To avoid sending duplicate redirects, use the **timeout** keyword and the *seconds* argument to indicate when to expire a redirect entry created.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring NHRP Shortcuts on the FlexVPN Spoke

Perform this task to configure NHRP shortcuts on the tunnel interface on the FlexVPN spoke.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. Do one of the following:
   - **ip nhrp shortcut** *virtual-template-number*
   - **ipv6 nhrp shortcut** *virtual-template-number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 1`	Configures the FlexVPN client interface and enters interface configuration mode.

Step 4	Do one of the following:
	• **ip nhrp shortcut** *virtual-template-number*
	• **ipv6 nhrp shortcut** *virtual-template-number*
	Example:
	`Device(config-if)# ip nhrp shortcut 1`
	Example:
	`Device(config-if)# ipv6 nhrp shortcut 1`
		Enables NHRP shortcuts on the FlexVPN client tunnel interface. This is necessary to establish spoke-to-spoke tunnels. The virtual-template number specified in this configuration and the virtual-template number specified in the Configuring the Virtual Tunnel Interface on the FlexVPN Spoke, on page 2744 task must be same.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring the Virtual Tunnel Interface on the FlexVPN Spoke

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number* **type tunnel**
4. **ip unnumbered tunnel** *number*
5. Do one of the following:
   - **ip nhrp network-id** *number*
   - **ipv6 nhrp network-id** *number*
6. Do one of the following:
   - **ip nhrp shortcut** *virtual-template-number*
   - **ipv6 nhrp shortcut** *virtual-template-number*
7. **ip nhrp redirect** [**timeout** *seconds*]
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **type tunnel** **Example:** `Device(config)# interface virtual-template 1 type tunnel`	Creates a virtual template interface that can be configured and applied dynamically to create virtual access interfaces.
Step 4	**ip unnumbered tunnel** *number* **Example:** `Device(config-if)# ip unnumbered tunnel 1`	Assigns the IPv4 address of the FlexVPN tunnel interface to the virtual tunnel interface.

	Command or Action	Purpose

Step 5	Do one of the following:
	• **ip nhrp network-id** *number*
	• **ipv6 nhrp network-id** *number*
	Example:
	`Device(config-if)# ip nhrp network-id 1`
	Example:
	`Device(config-if)# ipv6 nhrp network-id 1`
		Enables NHRP on the interface.

Step 6	Do one of the following:
	• **ip nhrp shortcut** *virtual-template-number*
	• **ipv6 nhrp shortcut** *virtual-template-number*
	Example:
	`Device(config-if)# ip nhrp shortcut 1`
	Example:
	`Device(config-if)# ipv6 nhrp shortcut 1`
		Enables NHRP shortcut switching on an interface.
		Note The current virtual template number must be specified. The virtual template number must be same as configured on the FlexVPN client tunnel interface.

Step 7	**ip nhrp redirect** [**timeout** *seconds*] **Example:** `Device(config-if)# ip nhrp redirect`	Enables NHRP redirects on the virtual tunnel interface. This is useful when networks move from one spoke to another. • To avoid sending duplicate redirects, use the **timeout** keyword and the *seconds* argument to indicate when to expire a redirect entry created.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Verifying the FlexVPN Spoke Configuration

Use the following commands to verify the FlexVPN spoke configuration.

SUMMARY STEPS

1. **show crypto ikev2 client flexvpn**
2. **show ipv6 route**
3. **show ipv6 nhrp**

DETAILED STEPS

Step 1 **show crypto ikev2 client flexvpn**

Example:

```
Device# show crypto ikev2 client flexvpn
```

```
  Profile : flexblk
  Current state:ACTIVE
  Peer : 4001::2000:1
  Source : Ethernet0/0
  ivrf : IP DEFAULT
  fvrf : IP DEFAULT
  Backup group: None
  Tunnel interface : Tunnel0
```

Displays the FlexVPN connection status between the FlexVPN server and client.

Step 2 **show ipv6 route**

Example:

```
Device# show ipv6 route

IPv6 Routing Table - default - 15 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   3001::/112 [0/0]
     via Tunnel0, directly connected
S   3001::1/128 [2/0], tag 1
     via 3001::1, Virtual-Access1 [Shortcut]
     via Virtual-Access1, directly connected
L   3001::2/128 [0/0]
     via Tunnel0, receive
S   3001::3/128 [2/0], tag 1
     via Tunnel0, directly connected
C   4001::2000:0/112 [0/0]
     via Ethernet0/0, directly connected
L   4001::2000:3/128 [0/0]
     via Ethernet0/0, receive
S   5001::/64 [2/0], tag 1
     via Tunnel0, directly connected
C   5001::2000:0/112 [0/0]
     via Loopback0, directly connected
L   5001::2000:1/128 [0/0]
     via Loopback0, receive
D   5001::3000:0/112 [90/28288000]
     via FE80::A8BB:CCFF:FE01:F400, Tunnel0
D   5001::4000:0/112 [90/28288000]
     via FE80::A8BB:CCFF:FE01:F400, Tunnel0
H   5001::4000:1/128 [250/1]
     via 3001::1, Virtual-Access1
C   5001::5000:0/112 [0/0]
     via Loopback1, directly connected
L   5001::5000:1/128 [0/0]
     via Loopback1, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Displays the IPv6 routes and Next Hop Resolution Protocol (NHRP) mapping information.

Step 3 **show ipv6 nhrp**

Example:

```
Device# show ipv6 nhrp

3001::1/128 via 3001::1
   Virtual-Access1 created 00:01:52, expire 01:58:14
   Type: dynamic, Flags: router implicit rib nho
   NBMA address: 172.17.1.9
    (Claimed NBMA address: 172.16.2.1)
5001::4000:1/128 via 3001::1
   Virtual-Access1 created 00:00:56, expire 01:59:03
   Type: dynamic, Flags: router rib
   NBMA address: 172.17.1.9
    (Claimed NBMA address: 172.16.2.1)
5001::5000:1/128 via 3001::2
   Virtual-Access1 created 00:01:52, expire 01:58:14
   Type: dynamic, Flags: router unique local
   NBMA address: 172.17.2.10
```

Example:

```
Device# show ipv6 nhrp

3001::1/128 via 3001::1
   Virtual-Access1 created 00:01:52, expire 01:58:14
   Type: dynamic, Flags: router implicit rib nho
   NBMA address: 4001::2000:2
5001::4000:1/128 via 3001::1
   Virtual-Access1 created 00:00:56, expire 01:59:03
   Type: dynamic, Flags: router rib
   NBMA address: 4001::2000:2
5001::5000:1/128 via 3001::2
   Virtual-Access1 created 00:01:52, expire 01:58:14
   Type: dynamic, Flags: router unique local
   NBMA address: 4001::2000:3
```

Displays the NHRP cache entries. In the first example, the output indicates that the transport is IPv4 (NBMA address). The remote spoke is behind Network Address Translation (NAT), as indicated by the Claimed NBMA address field, which is the pre-NAT address of the remote spoke. The cache entries also show the flags associated with each spoke, indicating the kind of route that has been inserted for each entry in the routing table. Next-Hop-Override (NHO) indicates the shortcut route. The *rib* flag indicates addition of an NHRP H route for that cache entry. The second example indicates that the transport is IPv6 (NBMA address). The remote spoke is not behind NAT, as indicated by the absence of claimed address in the output.

Troubleshooting Tips for FlexVPN Spoke Configuration

Here are few tips for troubleshooting FlexVPN spoke configuration:

1. Verify the connection between the spokes.
2. Check the configuration on the client (spoke) and the server.
3. Check the reachability of the remote hosts behind the spokes.
4. Verify the routing protocol configuration that is used to advertise the routes.
5. Verify that IKEv2 and IPsec are configured properly.
6. Verify the NHRP shortcut configuration on the spoke and the redirect configuration on the server (hub).

Problem	Troubleshooting Tips
Spoke to hub connection is not created.	A connection may not be created due to the absence of virtual access interfaces created at the hub. • Check the connectivity between the hub and spoke. • Use the **show crypto session** command to check the state of security associations (SAs) on the hub and spoke. • If SAs are active (as displayed in the **show crypto session** command), verify the output of the **show crypto ikev2 client flexvpn** command on the state of FlexVPN on the spoke.

Problem	Troubleshooting Tips
Spoke to spoke tunnel is not created.	Traffic must flow from spoke to spoke via the hub to initiate a spoke to spoke tunnel. • Verify the hub configuration to check if NHRP redirect is enabled. • Verify the spoke configuration to check if NHRP shortcut is enabled. • Verify the configuration in the FlexVPN server (hub) by using the **show ip** [**ipv6**] **nhrp traffic** command whether the hub has sent a traffic indirection to the spoke. • Verify the spokes have received the traffic and sent a resolution request by using either the **show ip** [**ipv6**] **nhrp traffic** command. • Verify the presence of NHRP cache entries for remote host and spoke on either spoke by using the **show ip** [**ipv6**] **nhrp** command. • Use the **show ip** [**ipv6**] **nhrp traffic** command on the remote spoke to verify that the resolution request is received. • Use the **show crypto ikev2 sa** command and the **show crypto session** command to verify that the spoke has received the resolution request and initiated a crypto session. • Use the **show ip** [**ipv6**] **interface brief** command to check if the virtual-access interface is present on both spokes. • Use the **show ip** [**ipv6**] **nhrp traffic** command on the spokes to verify that the resolution reply has been sent, and received by the peer on the virtual-access interface. • Use the **show ip** [**ipv6**] **nhrp** command to verify that the complete NHRP cache entries are present for the remote host and on all the spokes. • Use the **show ip** [**ipv6**] **route** command to check for the presence of H routes and/or next-hop-override (NHO) routes.

Configuration Examples for FlexVPN Spoke to Spoke

Example: Configuring FlexVPN Spoke to Spoke with Static Routing

The following example shows how to configure FlexVPN spoke to spoke with IKE-propagated static routing on the FlexVPN server and the FlexVPN client. The following is the configuration on the FlexVPN server:

```
hostname hub
!
crypto ikev2 authorization policy default
 pool flex-pool
 def-domain cisco.com
 route set interface
 route set access-list flex-route
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn hub.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.255
!
interface Ethernet0/0
 ip address 10.0.0.100 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nhrp network-id 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
ip local pool flex-pool 172.16.0.1 172.16.0.254
!
ip access-list standard flex-route
 permit any
```

The following is the configuration on the first FlexVPN client:

```
hostname spoke1
!
crypto ikev2 authorization policy default
 route set interface
 route set access-list flex-route
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn spoke1.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Tunnel0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.100
```

```
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 ip address 10.0.0.110 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.110.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
ip access-list standard flex-route
 permit 192.168.110.0 0.0.0.255
```

The following is the configuration on the second FlexVPN client:

```
hostname spoke2
!
crypto ikev2 authorization policy default
 route set interface
 route set access-list flex-route
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn spoke2.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Tunnel0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.100
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 ip address 10.0.0.120 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.120.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
ip access-list standard flex-route
 permit 192.168.120.0 0.0.0.255
```

Example: Configuring FlexVPN Spoke to Spoke with Dynamic Routing using BGP

The following example shows how to configure FlexVPN spoke to spoke with dynamic routing, using BGP on the FlexVPN server (with dynamic neighbor discovery) and the FlexVPN client. The following is the configuration on the FlexVPN server:

```
hostname hub
!
crypto ikev2 authorization policy default
 pool flex-pool
 def-domain cisco.com
 route set interface
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn hub.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.255
!
interface Ethernet0/0
 ip address 10.0.0.100 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nhrp network-id 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
ip local pool flex-pool 172.16.0.1 172.16.0.254
!
router bgp 65100
 bgp router-id 10.0.0.100
 bgp log-neighbor-changes
 bgp listen range 172.16.0.0/24 peer-group spokes
 neighbor spokes peer-group
 neighbor spokes remote-as 65100
 neighbor spokes transport connection-mode passive
 neighbor spokes update-source Loopback0
 !
 address-family ipv4
  neighbor spokes activate
  neighbor spokes default-originate
  neighbor spokes prefix-list no-default in
 exit-address-family
!
ip prefix-list no-default seq 5 deny 0.0.0.0/0
ip prefix-list no-default seq 10 permit 0.0.0.0/0 le 32
```

The following is the configuration on the first FlexVPN client:

```
hostname spoke1
!
crypto ikev2 authorization policy default
 route set interface
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn spoke1.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Tunnel0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.100
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 ip address 10.0.0.110 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.110.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
router bgp 65100
 bgp router-id 10.0.0.110
 bgp log-neighbor-changes
 neighbor hubs peer-group
 neighbor hubs remote-as 65100
 neighbor hubs update-source Tunnel0
 neighbor 172.16.1.1 peer-group hubs
 !
 address-family ipv4
  network 192.168.110.0
  neighbor 172.16.1.1 activate
 exit-address-family
```

The following is the configuration on the second FlexVPN client:

```
hostname spoke2
!
crypto ikev2 authorization policy default
 route set interface
 route set access-list flex-route
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local fqdn spoke2.cisco.com
 authentication local rsa-sig
```

```
 authentication remote rsa-sig
 pki trustpoint CA
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Tunnel0
 ip address negotiated
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.100
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 ip address 10.0.0.120 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.120.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Tunnel0
 ip nhrp network-id 1
 ip nhrp shortcut virtual-template 1
 ip nhrp redirect
 tunnel protection ipsec profile default
!
router bgp 65100
 bgp router-id 10.0.0.120
 bgp log-neighbor-changes
 neighbor hubs peer-group
 neighbor hubs remote-as 65100
 neighbor hubs update-source Tunnel0
 neighbor 172.16.1.1 peer-group hubs
 !
 address-family ipv4
  network 192.168.120.0
  neighbor 172.16.1.1 activate
 exit-address-family
```

Additional References for Configuring FlexVPN Spoke to Spoke

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*

Related Topic	Document Title
Shortcut Switching Enhancements	*Shortcut Switching Enhancements for NHRP in DMVPN Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for FlexVPN Spoke to Spoke

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 278: Feature Information for FlexVPN Spoke to Spoke

Feature Name	Releases	Feature Information
FlexVPN Spoke to Spoke		The FlexVPN Spoke to Spoke feature enables a FlexVPN client to establish a direct crypto channel with another FlexVPN client. The feature leverages VTIs, IKEv2, and NHRP to build spoke to spoke connections. In Cisco IOS Release 15.2(2)T, this feature was introduced. The following commands were introduced or modified: **ip unnumbered loopback0, tunnel source, tunnel mode gre ip, nhrp network-id, ip nhrp redirect, ip nhrp shortcut.**
FlexVPN IPv6 Direct Spoke to Spoke		The FlexVPN IPv6 Direct Spoke to Spoke feature supports the use of IPv6 addresses for FlexVPN spokes. The support for IPv6 addresses provides support for IPv6 over IPv4, IPv4 over IPv6, and IPv6 over IPv6 transports. The following commands were introduced or modified: **ipv6 nhrp shortcut.**

CHAPTER 210

Configuring IKEv2 Load Balancer

The IKEv2 Load Balancer feature provides support for enabling clusters of FlexVPN gateways and distributes incoming Internet Key Exchange Version 2 (IKEv2) connection requests among FlexVPN gateways. This feature redirects the incoming FlexVPN or AnyConnect client requests to the least loaded FlexVPN gateway based on the system and crypto load factors.

Prerequisites for IKEv2 Load Balancer

- For the server-side configuration, the Hot Standby Router Protocol (HSRP) and FlexVPN server (IKEv2 profile) must be configured.
- For the client-side configuration, the FlexVPN client must be configured.

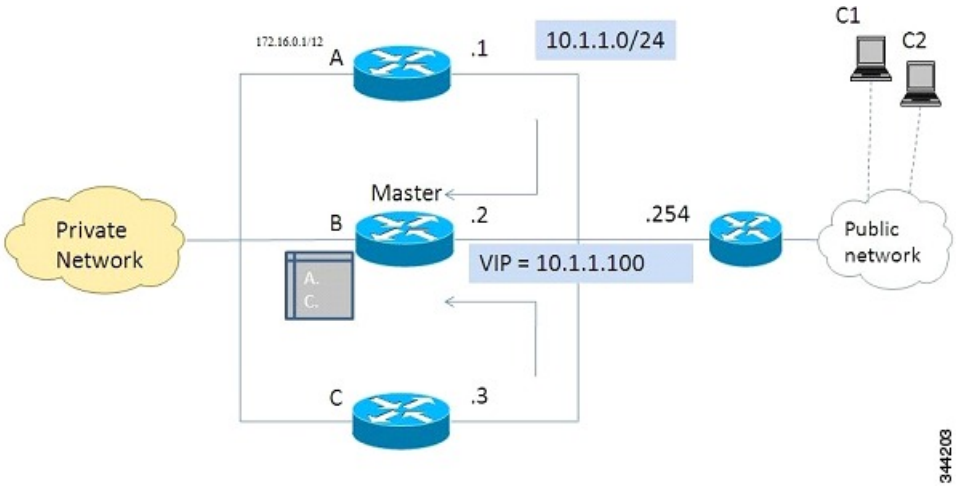
Information About IKEv2 Load Balancer

Overview of IKEv2 Load Balancer

The IKEv2 Load Balancer Support feature provides a Cluster Load Balancing (CLB) solution by redirecting requests from remote access clients to the Least Loaded Gateway (LLG) in the Hot Standby Router Protocol (HSRP) group or cluster. An HSRP cluster is a group of gateways or FlexVPN servers in a LAN or in an enterprise network. The CLB solution works with the Internet Key Exchange Version 2 (IKEv2) redirect mechanism defined in RFC 5685 by redirecting requests to the LLG in the HSRP cluster.

The figure below shows the working of the IKEv2 cluster load balancing solution.

Figure 109: IKEv2 Cluster Load Balancing Solution



1. An active HSRP gateway is elected as "primary" in the HSRP group and takes ownership of the Virtual IP address (VIP) for the group. The primary maintains a list of gateways in the cluster, keeps track of the load on each gateway, and redirects the FlexVPN client requests to the LLG.
2. The remaining gateways, termed as "subordinates," send load updates to the primary at periodic intervals.
3. When an IKEv2 client connects to the HSRP VIP, the request first reaches the primary, which in turn, redirects the request to the LLG in the cluster.

The components of the CLB solution are as follows:

- HSRP
- CLB primary
- CLB subordinate
- CLB communication
- IKEv2 redirects mechanism

Hot Standby Router Protocol

Hot Standby Router Protocol (HSRP) is used to elect the primary HSRP or Active Router (AR). For HSRP to elect a designated device, you must configure the VIP for one device in the group. This address is learned by other devices in the group. The IP address that is assigned to the primary is used as the VIP for the group. The HSRP active router (also called primary CLB) receives the IKEv2 requests and redirects these requests to the LLG in the cluster. The redirection is performed at the IKEv2 protocol level thereby achieving the following:

- All requests from the FlexVPN client reach the primary HSRP as the VIP is configured on the FlexVPN clients. The configuration of FlexVPN clients is minimized because the FlexVPN clients must only know the VIP of the HSRP cluster.
- The primary CLB is run on the same gateway as the primary HSRP, thereby maintaining the load information of all subordinate CLBs. The primary CLB enables effective redirection of requests and avoids multiple redirects and loops.

Primary CLB

A primary CLB runs on the primary HSRP or Active Router (AR). The primary receives updates from subordinate CLBs and sorts them based on their load condition to calculate the least loaded gateway (LLG). The primary sends the IP address of the LLG to IKEv2 (on the FlexVPN server). The IP address is sent to the initiator (FlexVPN client), which initiates an IKEv2 session with the LLG. The primary redirects incoming IKEv2 client connections towards the LLG. For more information, see IKEv2 Redirect Mechanism, on page 2759.

Note "CLB nodes" are used where both a primary CLB and CLB subordinate must be specified.

Subordinate CLB

A CLB subordinate runs on all devices in an HSRP group except on the Active Router (AR). The subordinates are responsible for sending periodic load updates to the server. A CLB subordinate is a fully functional IKEv2 gateway which supplies information to the primary CLB. Apart from updates, CLB subordinates send messages for aliveness assurance to the primary CLB.

CLB Load Management Mechanism

The CLB Load Management Mechanism is a TCP-based protocol that runs between the primary CLB and the CLB subordinates. The CLB load management mechanism informs the primary CLB about the load on the CLB subordinates. Based on this information, the primary CLB selects the LLG to handle the session on each new incoming IKEv2 connection.

Benefits of IKEv2 Load Balancer

- The IKEv2 Load Balancer Support feature is easy to configure and cost-effective.
- A FlexVPN client need not know the IP addresses of all gateways in the cluster. The client need only know the virtual IP address of the cluster.
- The entire crypto session is redirected to a node in the cluster.

IKEv2 Redirect Mechanism

The IKEv2 redirect mechanism enables a VPN gateway to redirect a FlexVPN client request to another VPN gateway based on load conditions and maintenance requirements.

The IKEv2 redirect mechanism is performed on security association (SA) initialization (IKE_SA_INIT) and on SA authentication (IKE_AUTH).

Redirect During IKEv2 Initial Exchange (SA Initialization)

A FlexVPN client, or an AnyConnect client indicates support for Internet Key Exchange Version 2 (IKEv2) redirect mechanism by including a REDIRECT_SUPPORTED notification message in the initial IKE_SA_INIT request. Use the **crypto ikev2 redirect client** command to enable the redirect mechanism on a client. Use the **crypto ikev2 redirect gateway init** command to enable redirect at IKE_SA_INIT on the gateway.

To redirect an IKEv2 request to another new gateway, the gateway that receives the IKE_SA_INIT request selects the IP address or the fully qualified domain name (FQDN) of the new gateway (in this case, the LLG) with help of the crypto load balancer (CLB) module. The gateway replies with an IKE_SA_INIT response that contains a REDIRECT notification message. The notification includes information such as the new gateway and the nonce value from the payload in the IKE_SA_INIT request. When a client receives the IKE_SA_INIT response, it verifies the nonce value sent in the IKE_SA_INIT request and the gateway information provided in the redirect notification, and confirms whether the redirect notification is as per the configuration.

Note If the nonce value does not match, the client discards the response and waits for another response, thereby preventing denial of service (DoS) attacks on the initiator. DoS attacks could be caused by an attacker injecting incorrect redirect payloads in IKE_SA_INIT responses.

In the IKE_SA_INIT exchange with the new gateway, the client message contains the REDIRECTED_FROM notification payload. The REDIRECTED_FROM notification payload consists of the IP address of the original VPN gateway that redirected the client. The IKEv2 exchange then proceeds as it would have proceeded with the original gateway.

Note The client may be redirected again by the new gateway if the new gateway also cannot serve the client. The client does not include the REDIRECT_SUPPORTED payload again in the IKE_SA_INIT exchange with the new gateway after the redirect. The presence of the REDIRECTED_FROM notification payload in the IKE_SA_INIT exchange with the new gateway indicates to the new gateway that the client supports the IKEv2 redirects mechanism.

Redirect During IKE_AUTH Exchange (SA Authentication)

A thorough security analysis shows that redirect during IKE_AUTH is neither more nor less secure than redirect during IKE_INIT. However, for performance and scalability reasons, we recommend redirect during IKE_INIT. Use the **crypto ikev2 redirect gateway auth** command to enable the redirect mechanism on the gateway. Use the **redirect gateway auth** command to enable redirect on authentication for selected IKEv2 profiles.

In this method, the client authorization payload is verified before sending the redirect notification payload. A client also verifies the gateway authorization payload before acting on the redirect notification. As the authorization payload is exchanged and successfully verified, the IKEv2 security association (SA) is validated successfully and the INITIAL_CONTACT is processed to decide on redirecting the request. If there is a redirect, the gateway creates the IKE SA and sends the IKE_AUTH response with the redirect notification.

A child SA is not created in this method. The IKE_AUTH does not contain a payload pertaining to a child SA. When the client receives the IKE_AUTH response, the client verifies the gateway authentication payload and deletes the IKEv2 SA with the gateway by sending a delete notification. The client acts on the redirect notification payload to establish connection with the new gateway. The client does not wait for an acknowledgment for the delete notification before establishing a connection with the new gateway. If the IKE_AUTH exchange involves the Extensible Authentication Protocol (EAP) authentication, the gateway has the choice of sending the redirect payload in the first or last IKE_AUTH response. The EAP authentication is included in the first IKE_AUTH response because it is not necessary to provide credentials for each redirect.

Compatibility and Interoperability

The IKEv2 redirect mechanism is based on RFC 5685. The gateway (IKEv2 responder) is compatible with clients (IKEv2 initiator) that implement the standard. Similarly, the client (initiator) implementation must be compatible with third party servers (responder) implementing the standard. The load management mechanism is Cisco proprietary and is only supported on Cisco IOS devices.

Handling Redirect Loops

A client request could be redirected multiple times in a sequence because of either an incorrect configuration or a denial of service (DoS) attack. In some cases, a client could enter a loop with two or more gateways redirecting the client to the other gateway thereby denying service to the client. To prevent this, a client can be configured, using the **crypto ikev2 redirect client** command with the **max-redirects** *number* keyword argument pair, to not accept more than a specific number of redirects for a particular IKEv2 security association (SA) setup.

IKEv2 Cluster Reconnect

The IKEv2 cluster reconnect feature allows Cisco AnyConnect client to reconnect to any server in the cluster. The **crypto ikev2 reconnect key** is introduced on the server to encrypt the opaque data pushed to the client. During failure detection, the client does reconnect with new or existing server without having to prompt for authentication credentials again.

There are only two key index values, 1 and 2 and at any point in time, any one of the keys configured using this will be active. The Cisco IOS server will be able to decrypt the reconnect data as long as the reconnect key is configured using the reconnect key CLI on the IOS server. This is true even if the key is only the back-up key.

This feature does not support when the **anyconnect-eap** keyword as authentication method in the IKEv2 profile through the **authentication** command.

Note This feature is available on Cisco IOS devices configured to work as Cisco AnyConnect server. The AnyConnect client software version that supports this feature are 4.2 and future releases. This feature is applicable for new deployments only. Once this feature is enabled on the Cisco IOS server, older releases of Cisco AnyConnect clients will not be supported.

How to Configure IKEv2 Load Balancer

Configuring the Server Cluster

Configuring an HSRP Group for Load Balancing

Perform this task to configure a single Hot Standby Router Protocol (HSRP) group for a cluster.

Hot Standby Router Protocol (HSRP) is used to elect the primary HSRP or Active Router (AR). For HSRP to elect a designated device, you must configure the VIP for one device in the group. This address is learned by other devices in the group. The IP address that is assigned to the primary is used as the VIP for the group.

The HSRP active router (also called primary CLB) receives the IKEv2 requests and redirects these requests to the LLG in the cluster. The redirection is performed at the IKEv2 protocol level thereby achieving the following:

- All requests from the FlexVPN client reach the primary HSRP as the VIP is configured on the FlexVPN clients. The configuration of FlexVPN clients is minimized because the FlexVPN clients must only know the VIP of the HSRP cluster.
- The primary CLB is run on the same gateway as the primary HSRP, thereby maintaining the load information of all CLB subordinates. The CLB primary enables effective redirection of requests and avoids multiple redirects and loops.

Note This task describes the minimum commands required to configure an HSRP group for load balancing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** *group-name*
7. **exit**
8. Repeat Steps 3 to 7 to configure an HSRP group for another cluster.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures the HSRP priority.
Step 6	**standby** *group-name* **Example:** `Device(config-if)# standby group1`	Specifies the name of the HSRP standby group.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 to 7 to configure an HSRP group for another cluster.	—

Configuring the Load Management Mechanism

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 cluster**
4. **holdtime** *milliseconds*
5. **master** {**overload-limit** *percent* | **weight** {**crypto-load** *weight-number* | **system-load** *weight-number*}}
6. **port** *port-number*
7. **slave** {**hello** *milliseconds* | **max-session** *number* | **priority** *number* | **update** *milliseconds*}
8. **standby-group** *group-name*
9. **shutdown**
10. **exit**
11. **crypto ikev2 reconnect key** *key index* **active** *name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ikev2 cluster** **Example:** `Device(config)# crypto ikev2 cluster`	Defines an IKEv2 cluster policy and enters IKEv2 cluster configuration mode.
Step 4	**holdtime** *milliseconds* **Example:** `Device(config-ikev2-cluster)# holdtime 10000`	(Optional) Specifies the time, in milliseconds, to receive messages from a peer. • If no messages are received within the configured time, the peer is declared "dead."
Step 5	**master** {**overload-limit** *percent* | **weight** {**crypto-load** *weight-number* | **system-load** *weight-number*}} **Example:** `Device(config-ikev2-cluster)# master weight crypto-load 10`	Specifies settings for the primary in the HSRP cluster. • **overload-limit** *percent*—The threshold load of the cluster. The load limit to decide when a device is busy and to ignore it when redirecting it for requests. • **weight**—Specifies the weight of a load attribute. Range: 0 to 100. Default: 100. • **crypto-load** *weight-number*—The IKE and IPsec security association (SA) load. • **system-load** *weight-number*—The system and memory load.
Step 6	**port** *port-number* **Example:** `Device(config-ikev2-cluster)# port 2000`	(Optional) Specifies the cluster primary listen port.
Step 7	**slave** {**hello** *milliseconds* | **max-session** *number* | **priority** *number* | **update** *milliseconds*} **Example:** `Device(config-ikev2-cluster)# slave max-session 90`	Specifies settings for subordinate gateways in the HSRP group. • **hello** *milliseconds*—The hello interval, in milliseconds, for a subordinate gateway. • **max-session** *number*—The maximum number of SAs allowed on a subordinate. This keyword is mandatory and cannot be skipped. • **priority** *number*—The subordinate priority. • **update** *milliseconds*—The interval, in milliseconds, between two update messages for a subordinate gateway.
Step 8	**standby-group** *group-name* **Example:** `Device(config-ikev2-cluster)# standby-group group1`	Defines the HSRP group containing the subordinates. • *group-name*—The group name is derived from the *group-name* argument specified in the **standby name** command.

	Command or Action	Purpose
Step 9	**shutdown** **Example:** `Device(config-ikev2-cluster)# shutdown`	(Optional) Disables the IKEv2 cluster policy.
Step 10	**exit** **Example:** `Device(config-ikev2-cluster)# exit`	Exits IKEv2 cluster configuration mode and returns to global configuration mode.
Step 11	**crypto ikev2 reconnect key** *key index* **active** *name* **Example:** `Device(config)# crypto ikev2 reconnect key 1 active test123`	Enables the IKEv2 opaque data support for session reconnect. **Note** The ikev2 cluster reconnect feature is enabled for encryption only when the **active** keyword is present in the **ikev2 reconnect key active** *name key-string*. The **active** keyword is mandatory to enable the cluster reconnect feature. If you use the **ikev2 reconnect key** *key-name key-string* command without the **active** keyword in the command, the headend will only be able to decrypt.
Step 12	**end** **Example:** `Device(config-ikev2-cluster)# end`	Exits IKEv2 cluster configuration mode and returns to privileged EXEC mode.

Activating the IKEv2 Redirect Mechanism on the Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 redirect gateway init**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ikev2 redirect gateway init** **Example:** `Device(config)# crypto ikev2 redirect gateway init`	Enables the IKEv2 redirect mechanism on the gateway during SA initiation.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Activating the IKEv2 Redirect Mechanism on the Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 redirect client** [**max-redirects** *number*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 redirect client** [**max-redirects** *number*] **Example:** `Device(config)# crypto ikev2 redirect client max-redirects 15`	Enables the IKEv2 redirect mechanism on the FlexVPN client. • **max-redirects** *number*—(Optional) Specifies the maximum number of redirects that can be configured on the FlexVPN client for redirect loop detection.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for IKEv2 Load Balancer

Example: Configuring an HSRP Group for Load Balancing

The following example shows RouterA configured as the active router for an Hot Standby Router Protocol (HSRP) group with a priority of 110. The default priority level is 100. This HSRP group is assigned the group name of group1. The group name is referred in the cluster policy.

```
Device(config)# hostname RouterA
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby group1
Device(config-if)# end
```

Example: Configuring the Load Management Mechanism

The following example shows how to configure the load management mechanism in IKEv2:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 cluster
Device(config-ikev2-cluster)# holdtime 10000
Device(config-ikev2-cluster)# master crypto-load 10
Device(config-ikev2-cluster)# port 2000
Device(config-ikev2-cluster)# slave priority 90
Device(config-ikev2-cluster)# standby-group group1
Device(config-ikev2-cluster)# shutdown
Device(config-ikev2-cluster)# end
```

Example: Configuring the Redirect Mechanism

The following example shows how to enable the redirect mechanism on a client and during initiation on a gateway:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 redirect client
Device(config)# crypto ikev2 redirect gateway init
Device(config)# end
```

Example: Configuring the Cluster Reconnect Key

The following example shows how to enable the reconnect key on a server:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 reconnect key 1 active key
```

```
Device(config)# crypto ikev2 reconnect key 2 test
Device(config)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
HSRP configuration	*Configuring HSRP*
HSRP commands	Cisco IOS First Hop Redundancy Protocols Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5685	*Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IKEv2 Load Balancer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 279: Feature Information for IKEv2 Load Balancer

Feature Name	Releases	Feature Information
IKEv2 fast convergence with cluster reconnect for Anyconnect		The IKEv2 fast convergence with cluster reconnect for Anyconnect feature enables the Cisco AnyConnect client to reconnect to any server in the cluster. The following command was introduced or modified: **crypto ikev2 reconnect key**
IKEv2 Load Balancer Support		The IKEv2 Load Balancer Support feature distributes incoming IKEv2 requests from FlexVPN clients among IKEv2 FlexVPN servers or gateways by redirecting requests to the least loaded gateway. The following commands were introduced or modified: **crypto ikev2 cluster, crypto ikev2 redirect, holdtime, primary (IKEv2), port (IKEv2), redirect gateway, subordinate (IKEv2), standby-group, show crypto ikev2 cluster, show crypto ikev2 sa.**

CHAPTER 211

Configuring IKEv2 Fragmentation

The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document.

Information About Configuring IKEv2 Fragmentation

IKEv2 Fragmentation

The Internet Key Exchange Version 2 (IKEv2) fragmentation protocol splits large IKEv2 message into a set of smaller ones, called IKE Fragment Messages. The IKEv2 fragmentation methodology, implemented on Cisco IOS software through the IKEv2 Remote Access Headend feature, is a Cisco proprietary method, which restricts interoperability with non-Cisco peers. The fragmentation is performed only on an encrypted IKEv2 packet, and hence, a peer cannot decrypt or authenticate the message until the peer receives all fragments. The IKE Fragmentation adhering to RFC feature implements the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document by encrypting packets after fragmentation, enabling interoperability with non-Cisco peers while continuing to support the Cisco proprietary fragmentation method.

Note By default, IKEv2 fragmentation is disabled, though show run all shows crypto ikev2 fragmentation mtu is 576 B.

Negotiation Between Peers

Effective with the IKE Fragmentation adhering to RFC feature, the support for the IETF standard fragmentation method is added the IKE_SA_INIT message as a notify payload, while Cisco proprietary Fragmentation method continues to be indicated using the Vendor ID payload in the same IKE_SA_INIT message. When fragmentation is enabled, support for both methodologies is displayed as appropriate in the **show crypto ikev2 sa detail** command. The maximum transmission unit (MTU) is configured locally and is not negotiated or

exchanged along with the messages. After the INIT exchange, the peers in a network configured with either methodology are aware of the authentication method that must be used and whether the AUTH message can be fragmented.

The following is a sample output from device when debug is enabled showing capability negotiation in INIT request message.

```
*Oct 14 08:45:24.732: IKEv2:(SESSION ID = 0,SA ID = 1):Next payload: SA, version: 2.0
Exchange type: IKE_SA_INIT, flags: INITIATOR Message id: 0, length: 524
Payload contents:
SA Next payload: KE, reserved: 0x0, length: 144
…
Security protocol id: IKE, spi size: 0, type: NAT_DETECTION_DESTINATION_IP
NOTIFY(IKEV2_FRAGMENTATION_SUPPORTED) Next payload: VID, reserved: 0x0, length: 8
Security protocol id: Unknown - 0, spi size: 0, type: IKEV2_FRAGMENTATION_SUPPORTED
VID Next payload: NONE, reserved: 0x0, length: 20
```

In the above output, the INIT request contains the intiator's message to a responder indicating support for both IETF standard fragmentation method and Cisco proprietary fragmentation method through the IKEV2_FRAGMENTATION_SUPPORTED and VID values in the message.

The following is a sample output from device when debug is enabled showing capability negotiation in INIT response message.

```
*Oct 14 08:45:24.732: IKEv2:(SESSION ID = 0,SA ID = 1):Next payload: SA, version: 2.0
Exchange type: IKE_SA_INIT, flags: INITIATOR Message id: 0, length: 524
Payload contents:
SA Next payload: KE, reserved: 0x0, length: 144
last proposal: 0x0, reserved: 0x0, length: 140
…
NOTIFY(IKEV2_FRAGMENTATION_SUPPORTED) Next payload: VID, reserved: 0x0, length: 8
Security protocol id: Unknown - 0, spi size: 0, type: IKEV2_FRAGMENTATION_SUPPORTED <--------
 Response, supporting both
VID Next payload: NONE, reserved: 0x0, length: 20 <-------- Response, supporting both
```

In the above output, the response request contains the responder's message to the initiator indicating support for both IETF standard fragmentation method and Cisco proprietary fragmentation method through the IKEV2_FRAGMENTATION_SUPPORTED and VID values in the message.

Fragmentation Support for Older Releases

To ensure fragmentation support for older releases having Cisco proprietary fragmentation method, IKEv2 continues to use the Vendor ID along with the IKEv2 notification payload type for the IETF standard fragmentation method. If both fragmentation methods are supported, IKEv2 prefers the IETF standard fragmentation method.

The following table indicates how the fragmentation type is determined based on the capability of peers. CISCO refers to Cisco proprietary fragmentation method and STD refers to the IETF standard fragmentation method.

Peer 1 Capability	Peer 2 Capability	Active Fragmentation Type on the Security Association
STD + CISCO	STD + CISCO	STD
STD	STD	STD
CISCO	CISCO	CISCO

Peer 1 Capability	Peer 2 Capability	Active Fragmentation Type on the Security Association
CISCO	STD + CISCO	CISCO
STD	STD + CISCO	STD
STD	CISCO	None
None	None or STD + CISCO or STD or CISCO	None

Encryption, Decryption, and Retransmission of Fragments

Fragmentation and Encryption

A packet is fragmented either based on the maximum transmission unit (MTU) value specified in the **crypto ikev2 fragmentation** command or the default MTU value. IKE messages that only contain the encrypted payload are fragmented. A new payload type—Encrypted and Authenticated Fragment—in the announcement message indicates the fragment number out of the total fragments. This payload is annotated as SKF and the value is 53.

Before the outgoing packet is encrypted, the packet length is checked. The security association established is also verified if the SA is enabled with the IETF standard fragmentation method. The following is a sample output from device displaying the transmission of fragmented packets.

```
*Oct 16 10:31:22.221: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 1 OF Total Fragments: 3
*Oct 16 10:31:22.222: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 2 OF Total Fragments: 3
*Oct 16 10:31:22.222: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 3 OF Total Fragments: 3
```

The line “SKF Next payload: COOP, reserved: 0x90, length: 216” and “SKF Fragment number: 1 OF Total Fragments: 3” indicate that the message is a Cooperative key server announcement (ANN) packet fragmented into three fragments.

Decryption and Defragmentation

When incoming fragments are received on a responder, each fragment is decrypted and stored temporarily. During defragmentation (assembling the fragments to the original pack), duplicate fragments, fragment numbers outside of total fragment number, and fragments having an entirely different fragment number are dropped. The fragments are added in ascending order of fragment number and not according to the received order), that way, packet assembly is faster. However, out of order fragments are allowed and processed. Each fragment is verified to ensure that all fragments that pertain to a message are received. If all fragments are

received, the packet is assembled from the fragments and processed as a newly received message. Acknowledgment (ACK) message is sent when the original packet is assembled, and not for each fragment.

Retransmissions

IKEv2 retransmissions happen as prompted by IKEv2 retransmission timers. The fragments once constructed and sent out for the first time, are held in a list, ready to be resent when the retransmission timers are triggered. When a retransmitted request is received, IKEv2 resends the response. The response is resent when the first fragment (#1) retransmission is received. The remaining fragment numbers are ignored, thereby allowing faster processing of the response.

Enabling Fragmentation

Use the **crypto ikev2 fragmentation** command to globally enable fragmentation per security association (SA). Fragmentation is enabled on SA when both peers indicate support for fragmentation after INIT exchange on each peers, to be used for IKE_AUTH exchange.

Note This command was introduced through IKEv2 Remote Access Headend feature and has not changed.

You can specify the maximum transmission unit (MTU), in bytes, using the **mtu** *mtu-size* keyword-argument pair. The MTU size refers to the IP or UDP encapsulated IKEv2 packets. The MTU range is from 68 to 1500 bytes. The default MTU size is 576 for IPv4 packets and 1280 bytes for IPv6 packets.

Effective with the IKE Fragmentation adhering to RFC feature, the **crypto ikev2 fragmentation** command:

- Affects future SAs only and does not affect the existing, old SAs.
- Supports Cisco proprietary fragmentation method and the IETF standard fragmentation method.

The **show crypto ikev2 sa detail** command displays the following information:

- The fragmentation method enabled on the peer. If the enabled fragmentation method is IETF standard fragmentation, the output displays the MTU, which is in use.
- Whether fragmentation is enabled on both peers or enabled on the local peer only.

IPv6 Support

The IKE Fragmentation adhering to RFC feature adds support for fragmenting IPv6 packets in IPv6 IKE endpoints when the IETF standard fragmentation method is used. The default MTU value is 1280 bytes and is used when the MTU is not specified in the **crypto ikev2 fragmentation** command. The MTU used in fragmentation is displayed in the output of the **show crypto ikev2 sa detail** command.

How to Configure Configuring IKEv2 Fragmentation

Configuring IKEv2 Fragmentation

Perform this task to enable automatic fragmentation of large IKEv2 packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 fragmentation** [**mtu** *mtu-size*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 fragmentation** [**mtu** *mtu-size*] **Example:** `Device(config)# crypto ikev2 fragmentation mtu 100`	Configures IKEv2 fragmentation. • The MTU range is from 96 to 1500 bytes. The default MTU size is 576 for IPv4 packets and 1280 bytes for IPv6 packets. **Note** The MTU size refers to the IP or UDP encapsulated IKEv2 packets.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Configuring IKEv2 Fragmentation

Example: IETF Fragmentation Enabled Displaying Configured MTU

The following is a sample output stating IETF standard fragmentation method is enabled. This statement is displayed when the responder supports IETF standard fragmentation method also. The output also displays the MTU in use.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none IN-NEG
Encr: Unknown - 0, PRF: Unknown - 0, Hash: None, DH Grp:0, Auth sign: Unknown - 0, Auth
verify: Unknown - 0
Life/Active Time: 86400/0 sec
```

```
CE id: 0, Session-id: 0
Status Description: Initiator waiting for INIT response
Local spi: 2CD1BEADB7C20854 Remote spi: 0000000000000000
Local id: 10.0.8.3
Remote id:
Local req msg id: 0 Remote req msg id: 0
Local next msg id: 1 Remote next msg id: 0
Local req queued: 0 Remote req queued: 0
Local window: 1 Remote window: 1
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 272 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

IPv6 Crypto IKEv2 SA
```

Example: IETF Standard Fragmentation Method Configured on the Initiator

The following is a sample output displaying IETF standard fragmentation method configured on the initiator, and the responder supports Cisco proprietary fragmentation method.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/59 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 84350219051DB9E3 Remote spi: 52A8BB3898E8B5CF
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 4 Remote req msg id: 0
Local next msg id: 4 Remote next msg id: 0
Local req queued: 4 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

IPv6 Crypto IKEv2 SA
```

The following is a sample output displaying the responder's configuration. Note that the output displays Cisco proprietary fragmentation method as configured, not enabled.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.9.4/848 10.0.8.3/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
```

```
verify: PSK
Life/Active Time: 86400/52 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 52A8BB3898E8B5CF Remote spi: 84350219051DB9E3
Local id: 10.0.9.4
Remote id: 10.0.8.3
Local req msg id: 0 Remote req msg id: 4
Local next msg id: 0 Remote next msg id: 4
Local req queued: 0 Remote req queued: 4
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No

IPv6 Crypto IKEv2 SA
```

The following example displays that the initiator supports IETF standard fragmentation method, whereas the responder does not support fragmentation. Note that the output states IETF standard fragmentation method is configured and not enabled.

```
Device# show crypto ikev2 sa detail
IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/44 sec
CE id: 1004, Session-id: 2
Status Description: Negotiation done
Local spi: 03534703287D9CA1 Remote spi: 146E1CFA68008A92
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 4 Remote req msg id: 0
Local next msg id: 4 Remote next msg id: 0
Local req queued: 4 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes
```

The following is a sample output displaying the responder's configuration. Note the statement "Fragmentation not configured."

```
Device# show crypto ikev2 sa detail
IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.9.4/848 10.0.8.3/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/23 sec
CE id: 1004, Session-id: 4
Status Description: Negotiation done
Local spi: 146E1CFA68008A92 Remote spi: 03534703287D9CA1
Local id: 10.0.9.4
Remote id: 10.0.8.3
```

```
Local req msg id: 0 Remote req msg id: 3
Local next msg id: 0 Remote next msg id: 3
Local req queued: 0 Remote req queued: 3
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No
```

Example: IETF Standard Fragmentation Method not Configured on the Initiator

The following is a sample output displaying no fragmentation method configured on the initiator.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
2 10.0.8.3/848 10.0.9.4/848 none/none DELETE
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/28 sec
CE id: 1001, Session-id: 1
Status Description: Deleting IKE SA
Local spi: 1A375C00C1D157CF Remote spi: DB50F1BC58814FFA
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 2 Remote req msg id: 4
Local next msg id: 4 Remote next msg id: 5
Local req queued: 2 Remote req queued: 4
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No

IPv6 Crypto IKEv2 SA
```

Example: IPv6 Support for Fragmentation

This following example shows fragmentation on FlexVPN endpoints—hub and spoke. The following configuration pertains to the hub, which is configured with a maximum transmission unit (MTU) of 1300 for fragmenting the packets.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

IPv6 Crypto IKEv2 SA

Tunnel-id fvrf/ivrf Status
1 none/none READY
Local 4001::2000:3/500
Remote 4001::2000:1/500
```

```
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/64 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 45BA0D30D0EB5FFF Remote spi: 8D7B5A8389CEB8B3
Local id: R2.cisco.com
Remote id: R1.cisco.com
Local req msg id: 3 Remote req msg id: 0
Local next msg id: 3 Remote next msg id: 0
Local req queued: 3 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 1272 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes
Remote subnets:
10.0.0.251 255.255.255.255
IPv6 Remote subnets:
3001::/112
5001::/64
```

The following configuration pertains to the spoke, which is configured with the default MTU.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

IPv6 Crypto IKEv2 SA

Tunnel-id fvrf/ivrf Status
1 none/none READY
Local 4001::2000:1/500
Remote 4001::2000:3/500
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/58 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 8D7B5A8389CEB8B3 Remote spi: 45BA0D30D0EB5FFF
Local id: R1.cisco.com
Remote id: R2.cisco.com
Local req msg id: 0 Remote req msg id: 3
Local next msg id: 0 Remote next msg id: 3
Local req queued: 0 Remote req queued: 3
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 1232 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No
Remote subnets:
10.0.0.3 255.255.255.255
```

Additional References for Configuring IKEv2 Fragmentation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security Commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Standards and RFCs

Standard/RFC	Title
IKEv2 Fragmentation	*draft-ietf-ipsecme-ikev2-fragmentation-10*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IKEv2 Fragmentation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 280: Feature Information for Configuring IKEv2 Fragmentation

Feature Name	Releases	Feature Information
IKEv2 Fragmentation adhering to RFC		The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document. The following command was modified: **show crypto ikev2 sa**.

CHAPTER 212

Configuring IKEv2 Reconnect

The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client.

Prerequisites for Configuring IKEv2 Reconnect

- You must enable the BypassDownloader function in the AnyConnectLocalPolicy file by setting the <BypassDownloader> value to true. If your device does not support SSL, the BypassDownloader function will not work.You must disable the function manually by setting the <BypassDownloader> value to false, else the connection will fail.

Restrictions for Configuring IKEv2 Reconnect

- The preshared key authorization method cannot be configured on the Internet Key Exchange Version 2 (IKEv2) profile. This is because the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature uses the preshared key authorization method and configuring the preshared key on the same IKEv2 profile may lead to confusion.
- The following commands cannot be configured on the IKEv2 profile: **authentication local pre-share**, **authentication remote pre-share**, **keyring**, **aaa authorization group psk**, and **aaa authorization user psk**.

Information About Configured IKEv2 Reconnect

IKEv2 and Cisco AnyConnect Client Reconnect Feature

The Auto Reconnect feature in the Cisco AnyConnect client helps the Cisco AnyConnect VPN client to remember the session for a period of time and to resume the connection after establishing the secure channel. As the Cisco AnyConnect Client is extensively used with Internet Key Exchange Version 2 (IKEv2), IKEv2 extends the Auto Reconnect feature support on Cisco IOS software through the IOS IKEv2 support for Auto Reconnect feature of AnyConnect feature.

Auto Reconnect in the Cisco AnyConnect client occurs in the following scenarios:

- The intermediate network is down. The Cisco AnyConnect client tries to resume the session when it is up.
- The Cisco AnyConnect client device switches between networks. This results in source IP or port change, which brings down the existing security association (SA) and, hence, the Cisco AnyConnect client tries to resume the SA using the Auto Reconnect feature.
- The Cisco AnyConnect client device tries to resume SA after returning from sleep or hibernate mode.

Advantages of Using the Auto Reconnect Feature

- The copy attributes used in the original session are reused without querying the authentication, authorization, and accounting (AAA) server.
- The Cisco IOS gateway does not have to contact the RADIUS server for reconnecting to the client.
- No user interaction for authentication or authorization is needed during resuming the session.
- The authentication method is the preshared key when reconnecting a session. This authentication method is quick compared to other authentication methods (that include Rivest, Shamir, and Adelman (RSA) signature authentication method, Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method, and Extensible Authentication Protocol (EAP) authentication method). The preshared key authentication method helps in resuming a session on the IOS software with minimal resources.
- The unused security associations (SAs) are removed thereby freeing the crypto resources.

Auto Reconnect and DPD

Dead Peer Detection (DPD) is configured to confirm the availability of a peer send by sending queries to a peer. If there are no responses from the peer, the security association created for that peer is deleted. You need not configure DPD in a reconnect profile if DPD configured on the FlexVPN server because in both configuration scenarios, the purpose is the same . However, if the feature is enabled, DPD is queued as on demand DPD in IKEv2, which also stores the platform specific handle when deleting the SA.

Message Exchanges Between Cisco IOS Gateway and Cisco AnyConnect Client

The Cisco AnyConnect client contacts the Cisco IOS gateway to establish a security association (SA). During authorization or AUTH exchange (CFGMODE_REQ payload of IKE_AUTH request), IKEv2 checks if the IOS IKEv2 support for the Auto Reconnect feature of AnyConnect feature is enabled in the IKEv2 profile using the **reconnect** command, selects the IKEv2 policy of the chosen IKEv2 profile, and sends the session

ID and the session token attributes to the Cisco AnyConnect client in CFGMODE_REPLY payload of the IKE_AUTH response. The authorization method is the preshared key between the client and Cisco IOS software for the SA.

IKEv2 periodically sends dead peer detection (DPD) messages to the Cisco AnyConnect client to validate if the client is active. The Cisco AnyConnect client responds to the DPD messages, which the Cisco IOS gateway understands as an active client and creates a security association (SA) with the client. However, if the client does not reconnect within 30 minutes, which is the default reconnect timeout period, the Cisco IOS gateway assumes that the client is inactive and deletes the SA for that client. The Cisco AnyConnect client needs to start a fresh connection.

Use the **show crypto ikev2 stats reconnect** command to view the connection statistics and the **clear crypto ikev2 session** command to delete the SA with the client.

How to Configure IKEv2 Reconnect

Enabling IKEv2 Reconnect

Perform this task to enable the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **reconnect** [**timeout** *seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**reconnect** [**timeout** *seconds*] **Example:** `Device(config-ikev2-profile)# reconnect timeout 900`	Enables the IKEv2 support for the Auto Reconnect feature.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Troubleshooting IKEv2 Reconnect Configuration

Use the following commands to verify or clear the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature configuration.

SUMMARY STEPS

1. **enable**
2. **show crypto ikev2 stats reconnect**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show crypto ikev2 stats reconnect**

Displays the reconnect statistics.

Example:

```
Device# show crypto ikev2 stats reconnect

Total incoming reconnect connection:        10
Success reconnect connection:               10
Failed reconnect connection:                0
Reconnect capable active session count:     4
Reconnect capable inactive session count:   6
```

Configuration Examples for Configuring IKEv2 Reconnect

Example: Enabling IKEv2 Reconnect

The following example shows how to enable the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature.

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 profile profile1
Device(config-ikev2-profile)# reconnect timeout 600
Device(config-ikev2-profile)# end
```

Additional References for Configuring IKEv2 Reconnect

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Cisco AnyConnect VPN Client Information	*Cisco AnyConnect VPN Client Administrator Guide, Release 2.4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IKEv2 Reconnect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 281: Feature Information for Configuring IKEv2 Reconnect

Feature Name	Releases	Feature Information
IOS IKEv2 support for AutoReconnect feature of AnyConnect		The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client. The following commands were introduced or modified: **clear crypto ikev2 stats, reconnect, show crypto ikev2 stats.**

CHAPTER 213

Configuring MPLS over FlexVPN

Last Published Date: March 28, 2014

The MPLS over FlexVPN feature implements Multiprotocol Label Switching (MPLS) over a dynamically established IPsec tunnel thereby supporting duplicate address spaces.

Prerequisites for MPLS over FlexVPN

- Internet Key Exchange Version 2 (IKEv2) and IPsec must be configured.
- MPLS must be configured.
- NHRP redirect must be configured.

Information About Configuring MPLS over FlexVPN

MPLS and FlexVPN

Network domains having overlapping addressing spaces use VPN routing and forwarding (VRF) to segregate traffic so that data intended for one domain does not enter another domain. Data security between the provider-edge (PE) devices is achieved by defining an tunnel interface with IPsec protection for every VRF. This ensures that traffic from every domain passes over the corresponding IPsec tunnel. However as the number of domains and nodes grow in a network, this may not be scalable because every protected domain requires a separate IPsec tunnel and an interface.

Multiprotocol Label Switching (MPLS) provides the ability to assign labels per VRF or per prefix, which identifies the correct VRF into which data needs to be routed to. This can be achieved with just a single MPLS-aware interface having IPsec protection and a single IPsec tunnel between the PEs.

The MPLS over FlexVPN feature provides a solution to achieve communication between overlapping addresses in customer networks when a remote customer network needs to be discovered dynamically using Next Hop

Resolution Protocol (NHRP) and at the same time secure the data traffic between the PE devices using IPsec. This solution can be used by customers who have deployed MPLS network and want to extend their MPLS network to a newly configured network (determined dynamically) in a different region over the Internet in a secure way.

The components of the MPLS over FlexVPN solution are as follows:

- IPsec—Secures the data traffic between the spoke and the hub and between the spokes after the remote spoke is discovered dynamically.
- Internet Key Exchange Version 2 (IKEv2)—Adds static routes to the peer's tunnel overlay address as a directly connected route. This route results in adding an implicit null label to the Label Information Base (LIB) for the peer's tunnel overlay address.

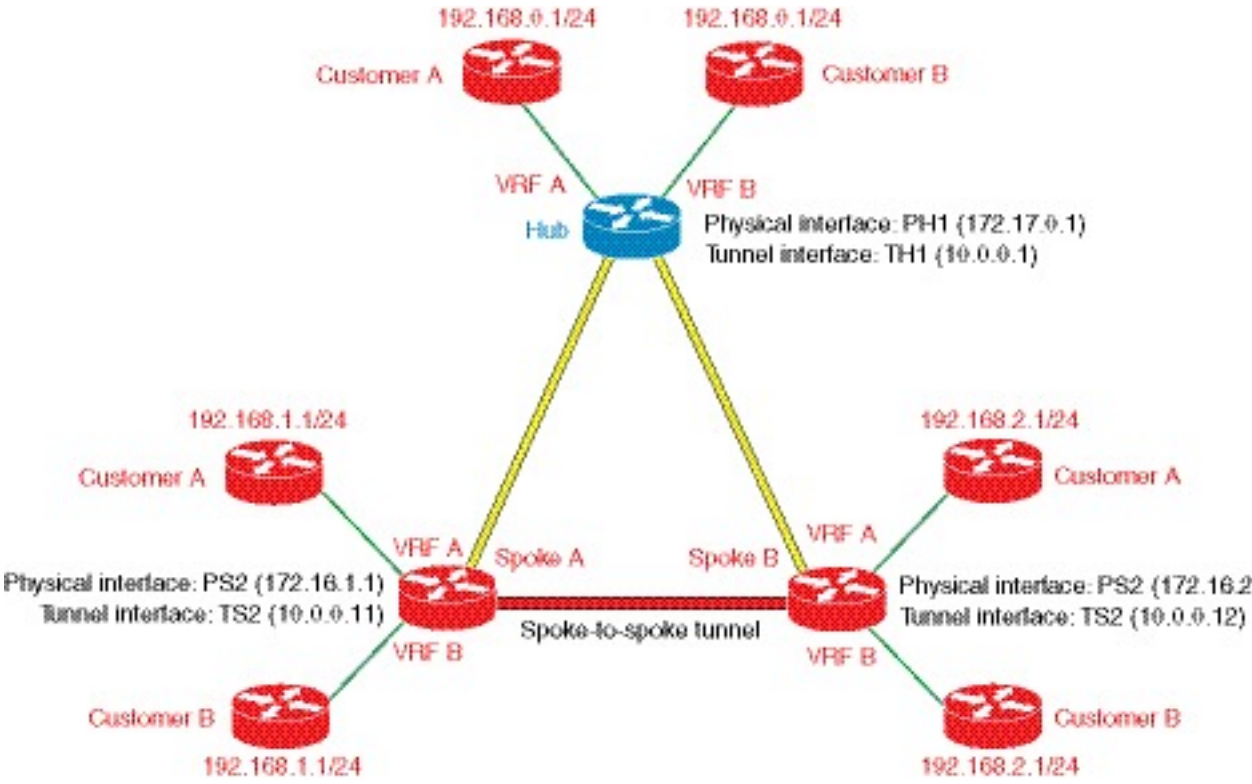
Note IKEv2 is used instead of LDP because LDP involves establishing TCP channel with every LDP neighbor. Enabling LDP keeps the spoke-to-spoke channel active due to the LDP hello traffic thereby never bringing down the spoke-to-spoke channel. Therefore, the **mpls ip** command must never be executed on the tunnel interface or virtual template when configuring the MPLS over FlexVPN feature.

- NHRP—Used to resolve the remote overlay address and dynamically discover the transport end point needed to establish a secure tunnel. If a multipoint generic routing encapsulation (GRE) interface is used, the tunnel end point database stores the mapping between the overlay and corresponding nonbroadcast multiaccess (NBMA) address.
- MPLS—Enables MPLS tag switching for data packets. By default, Label Distribution Protocol (LDP) is not enabled and is not enabled between the spokes because LDP keepalive will try to keep the spoke-spoke tunnel up and is not desired in the absence of data traffic.
- MPLS Forwarding Infrastructure (MFI)—Allocates and releases labels by the applications; NHRP is an application that call MFI for label management.
- Multiprotocol BGP (MP-BGP)—Distributes overlay labels for the network on different VRFs.

Working of MPLS over FlexVPN

The following figure along description explains the working of MPLS over FlexVPN solution:

Figure 110: Spoke to Hub to Spoke Topology



The MPLS over FlexVPN solution has the following assumptions:

- Multiprotocol BGP (MP-BGP) allows distributing labels per VPN routing and forwarding (VRF) or per prefix.
- Label 10 is assigned to VRF A for packets that arrive from hub to spoke A.
- Label 20 is assigned to VRF A for packets that arrive from the hub to spoke B.
- Label 30 is assigned to VRF A on the hub for packets that arrive from spoke A to the hub.
- Label 40 is assigned to VRF B on the hub for packets that arrive from spoke B to the hub.

1. IKEv2 and IPsec security associations are established from each spoke to the hub. IKEv2 installs implicit null label values for the spoke's overlay address that is received in the mode config reply and mode config set.

Note Implicit null label is installed since the spoke and hub are always next-hop to each other in the overlay space.

2. MP-BGP exchanges the label per VRF or label per prefix with all the VRFs.
3. After the labels and routes have been exchanged, data forwarding begins. When the first data packet destined for 192.168.2.1 arrives on spoke A on VRF A, it is forwarded to the hub. The packet is label encapsulated using generic routing encapsulation (GRE), only containing the overlay label, and encrypted.
4. The data packet is decrypted when it reaches the hub on the physical (virtual access) interface or the tunnel interface which is 172.17.0.1 and 10.0.0.1 respectively. The overlay label is looked up in the hub, the packet is encapsulated using GRE, encrypted and sent to spoke B.

5. An NHRP redirect packet is sent from the hub to spoke A. As label 30 identifies the VRF on which the data packet arrived, the VRF information is conveyed to NHRP.
6. NHRP processes the redirect packet and triggers an NHRP resolution request. An NHRP mapping entry is created and VRF A is associated for the prefix that needs to be resolved.
7. The resolution request is sent to the hub, which looks up its overlay label and sends the resolution request to the appropriate destination, which in this case is Spoke B.
8. NHRP resolution request arrives on Spoke B and creates a virtual access interface or an multipoint GRE (mGRE) interface on Spoke B.
9. An IKEv2 and IPsec session is initiated from Spoke B to Spoke A resulting in the creation of a virtual access interface or mGRE interface on Spoke A. NHRP adds the route for IP address of Spoke A tunnel via the newly created virtual access interface.
10. NHRP resolution reply from Spoke B carries the label value that may be used by Spoke A for sending data over the spoke-to-spoke tunnel. Therefore, NHRP allocates a label from the MPLS forwarding instance (MFI) and sends this label information to Spoke A to be used for the spoke-to-spoke tunnel.

Note MFI tracks the labels. If a label is already allocated and assigned to MP-BGP for a particular VRF, the label is returned to NHRP. MFI tracks the number of applications using this a particular label and returns the label back to pool only when all the applications have released the label.

11. NHRP resolution reply also contains an implicit null label for the IP address of the virtual access interface or mGRE interface on Spoke B. In this example, the reply would be 192.168.2.0/24, label 40, 10.0.0.12, 172.16.2.1, [implicit-NULL].
12. NHRP resolution reply is received at the virtual access interface or mGRE interface on Spoke A. The NHRP request ID present in reply packet is matched with the request ID of the request that was initially sent by Spoke A to know the VRF for which the request was sent. NHRP cache is looked up to find the NHRP entry and the entry is termed “Complete”. NHRP inserts a route into the VRF routing table with the label information.
13. Routes and labels are setup between Spoke A and Spoke B. Data is now label encapsulated and encrypted over the spoke-to-spoke dynamically established tunnel between Spoke A and Spoke B.

IVRF Support for FlexVPN

The Inside VPN Routing and Forwarding (IVRF) support for FlexVPN provides the capability of performing the following NHRP routing operations in the IVRF configured on the tunnel interface:

- Sending NHRP resolution request after performing the route lookup.
- Forwarding of NHRP resolution request on the hub.
- Creating an H route or next-hop override (NHO) in the IVRF when creating a shortcut tunnel
- Deleting the H route or NHO from the IVRF when the shortcut tunnel is deleted

How to Configure MPLS over FlexVPN

Configuring MPLS over FlexVPN

Perform this task to configure MPLS over FlexVPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **mpls nhrp**
5. **end**
6. **show mpls forwarding-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 1`	Configures the FlexVPN client interface and enters interface configuration mode.
Step 4	**mpls nhrp** **Example:** `Device(config-if)# mpls nhrp`	Enables MPLS tag switching without enabling Label Distribution Protocol (LDP).
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**show mpls forwarding-table** **Example:** `Device# show mpls forwarding-table`	Displays information about the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB).

Configuration Examples for Configuring MPLS over FlexVPN

Example: Configuring MPLS over FlexVPN

The following example shows how to transport multiple customer VRFs on FlexVPN leveraging MPLS functionality. The following is the configuration on spoke 1.

```
hostname R3-Spoke1
boot-start-marker
```

```
boot-end-marker
!
!
vrf definition cust1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
 exit-address-family
!
vrf definition cust2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
 !
 address-family ipv4
 exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
mpls ldp loop-detection
!
crypto pki trustpoint CA
 enrollment url http://172.16.1.1:80
 password
 fingerprint E0AFEFD7F08070BAB33C8297C97E6457
 subject-name cn=R3-spoke.cisco.com,OU=FLEX,O=Cisco
 revocation-check crl none
!
crypto pki certificate map mymap 10
 subject-name co ou = flex
!
crypto pki certificate chain CA
 certificate 03
 certificate ca 01
crypto ikev2 authorization policy default
 route set interface
!
crypto ikev2 profile default
 match certificate mymap
 identity local fqdn R3-Spoke.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 dpd 60 2 on-demand
 aaa authorization group cert list default default
!
!
!
!
crypto ipsec profile default
 set ikev2-profile default
!
!
!
!
!
!
interface Tunnel0
```

```
 ip address negotiated
 mpls bgp forwarding
 tunnel source Ethernet0/0
 tunnel destination 172.16.0.1
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 description WAN
 ip address 172.16.1.103 255.255.255.0
!
interface Ethernet0/1
 description LAN
 no ip address
 no ip unreachables
!
interface Ethernet0/1.10
 encapsulation dot1Q 10
 vrf forwarding cust1
 ip address 192.168.113.1 255.255.255.0
!
interface Ethernet0/1.20
 encapsulation dot1Q 20
 vrf forwarding cust2
 ip address 192.168.123.1 255.255.255.0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 10
 neighbor 10.0.0.1 ebgp-multihop 255
 neighbor 10.0.0.1 update-source Tunnel0
 !
 address-family ipv4
  neighbor 10.0.0.1 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf cust1
  redistribute connected
 exit-address-family
 !
 address-family ipv4 vrf cust2
  redistribute connected
 exit-address-family
!
ip route 10.0.0.1 255.255.255.255 Tunnel0 name workaround
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB
```

The following is spoke 2 configuration.

```
hostname R4-Spoke
!
vrf definition cust1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
 exit-address-family
!
vrf definition cust2
 rd 2:2
```

```
 route-target export 2:2
 route-target import 2:2
 !
 address-family ipv4
 exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
crypto pki token default removal timeout 0
!
crypto pki trustpoint CA
 enrollment url http://172.16.1.1:80
 password
 fingerprint E0AFEFD7F08070BAB33C8297C97E6457
 subject-name cn=R4-Spoke.cisco.com,OU=Flex,O=Cisco
 revocation-check crl none
!
crypto pki certificate map mymap 10
 subject-name co ou = flex
!
crypto pki certificate chain CA
 certificate 04
 certificate ca 01
!
crypto ikev2 authorization policy default
 route set interface
!
crypto ikev2 profile default
 match certificate mymap
 identity local fqdn R4.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 dpd 60 2 on-demand
 aaa authorization group cert list default default
 virtual-template 1
!
crypto ipsec profile default
 set ikev2-profile default
!
interface Loopback100
 vrf forwarding cust1
 ip address 192.168.114.1 255.255.255.0
!
interface Loopback101
 vrf forwarding cust2
 ip address 192.168.124.1 255.255.255.0
!
interface Tunnel0
 ip address negotiated
 mpls bgp forwarding
 tunnel source Ethernet0/0
 tunnel destination 172.16.0.1
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 description WAN
 ip address 172.16.1.104 255.255.255.0
!
```

```
interface Ethernet0/1
 description LAN
 ip address 192.168.104.1 255.255.255.0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 10
 neighbor 10.0.0.1 ebgp-multihop 255
 neighbor 10.0.0.1 update-source Tunnel0
 !
 address-family ipv4
  neighbor 10.0.0.1 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf cust1
  redistribute connected
 exit-address-family
 !
 address-family ipv4 vrf cust2
  redistribute connected
 exit-address-family
!
ip route 10.0.0.1 255.255.255.255 Tunnel0
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB
```

The following is the hub configuration.

```
hostname R1-HUB
aaa new-model
!
!
aaa authorization network default local
!
!
clock timezone CET 1 0
!
ip vrf cust1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
ip vrf cust2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp loop-detection
!
crypto pki trustpoint CA
 enrollment url http://172.16.0.2:80
 password
 fingerprint E0AFEFD7F08070BAB33C8297C97E6457
 subject-name CN=R1-HUB.cisco.com,OU=FLEX,OU=VPN,O=Cisco Systems,C=US,L=Linux
 revocation-check crl none
```

```
 rsakeypair R1-HUB.cisco.com 2048
 auto-enroll 95
!
!
crypto pki certificate chain CA
 certificate 02
 certificate ca 01
!
redundancy
!
!
!
crypto ikev2 authorization policy default
 pool mypool
 banner ^C Welcome ^C
 def-domain cisco.com
!
!
!
!
crypto ikev2 profile default
 match identity remote fqdn domain cisco.com
 identity local dn
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 dpd 60 2 on-demand
 aaa authorization group cert list default default
 virtual-template 1
!

crypto ipsec profile default
 set ikev2-profile default
!
!
!
!
!
!
interface Loopback0
 description VT source interface
 ip address 10.0.0.1 255.255.255.255
!
interface Ethernet0/0
 description WAN
 ip address 172.16.0.1 255.255.255.252
!
interface Ethernet0/1
 description LAN
 ip address 192.168.100.1 255.255.255.0
!
interface Ethernet0/2
 ip vrf forwarding cust1
 ip address 192.168.110.1 255.255.255.0
!
interface Ethernet0/3
 ip vrf forwarding cust2
 ip address 192.168.111.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nhrp network-id 1
 ip nhrp redirect
 mpls bgp forwarding
```

```
 tunnel protection ipsec profile default
!
router bgp 10
 bgp log-neighbor-changes
 bgp listen range 0.0.0.0/0 peer-group mpls
 bgp listen limit 5000
 neighbor mpls peer-group
 neighbor mpls remote-as 100
 neighbor mpls transport connection-mode passive
 neighbor mpls update-source Loopback0
 !
 address-family ipv4
  redistribute static route-map global
  neighbor mpls activate
  neighbor mpls next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor mpls activate
  neighbor mpls send-community both
 exit-address-family
 !
 address-family ipv4 vrf cust1
  redistribute connected
  redistribute static route-map cust1
  default-information originate
 exit-address-family
 !
 address-family ipv4 vrf cust2
  redistribute connected
  redistribute static route-map cust2
  default-information originate
 exit-address-family
!
ip local pool mypool 10.1.1.1 10.1.1.254
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 172.16.0.2 name route_to_internet
ip route vrf cust1 0.0.0.0 0.0.0.0 Null0 tag 666 name default_originate
ip route vrf cust2 0.0.0.0 0.0.0.0 Null0 tag 667 name default_originate
!
route-map cust1 permit 10
 match tag 666
!
route-map cust2 permit 10
 match tag 667
```

The following is sample output from the spoke.

```
Device# show ip cef vrf cust1 192.168.110.1

192.168.110.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
 sharing
  sources: RIB
  feature space:
   IPRM: 0x00018000
   LFD: 192.168.110.0/24 0 local labels
        contains path extension list
  ifnums: (none)
  path EF36CA28, path list EF36DEB4, share 1/1, type recursive, for IPv4, flags
must-be-labelled
    MPLS short path extensions: MOI flags = 0x0 label 19
```

```
  recursive via 10.0.0.1[IPv4:Default] label 19, fib F0C5926C, 1 terminal fib,
v4:Default:10.0.0.1/32
    path EF36CBE8, path list EF36DFF4, share 1/1, type attached host, for IPv4
      MPLS short path extensions: MOI flags = 0x1 label implicit-null
    attached to Tunnel0, adjacency IP midchain out of Tunnel0 F0481718
  output chain: label 19 label implicit-null TAG midchain out of Tunnel0 F1D97A90 IP adj
out of Ethernet0/0, addr 172.16.1.1 F0481848
R4-Spoke#sh ip bgp vpnv4 all label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (cust1)
   0.0.0.0          10.0.0.1        nolabel/18
   192.168.110.0    10.0.0.1        nolabel/19
   192.168.114.0    0.0.0.0         16/nolabel(cust1)
Route Distinguisher: 2:2 (cust2)
   0.0.0.0          10.0.0.1        nolabel/20
   192.168.111.0    10.0.0.1        nolabel/21
   192.168.124.0    0.0.0.0         17/nolabel(cust2)
```

The following is sample output from the hub.

```
Device# show ip cef vrf cust1 192.168.113.1

192.168.113.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
 sharing
  sources: RIB, LTE
  feature space:
   IPRM: 0x00018000
   LFD: 192.168.113.0/24 1 local label
   local label info: other/25
        contains path extension list
        disposition chain 0xF1E1D9B0
        label switch chain 0xF1E1D9B0
  ifnums: (none)
  path F16ECA10, path list F16EDFBC, share 1/1, type recursive, for IPv4, flags
must-be-labelled
    MPLS short path extensions: MOI flags = 0x0 label 16
  recursive via 10.1.1.3[IPv4:Default] label 16, fib F0CCD6E8, 1 terminal fib,
v4:Default:10.1.1.3/32
    path F16ECE00, path list F16EE28C, share 1/1, type attached host, for IPv4
      MPLS short path extensions: MOI flags = 0x1 label implicit-null
    attached to Virtual-Access1, adjacency IP midchain out of Virtual-Access1 F04F35D8
  output chain: label 16 label implicit-null TAG midchain out of Virtual-Access1 F1E1DF60
IP adj out of Ethernet0/0, addr 172.16.0.2 F04F3708
R1-HUB#sh ip bgp vpnv4 all
BGP table version is 49, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f
RT-Filter, a additional-path
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust1)
*> 0.0.0.0          0.0.0.0                  0         32768 ?
*> 192.168.110.0    0.0.0.0                  0         32768 ?
*> 192.168.113.0    10.1.1.3                 0             0 100 ?
*> 192.168.114.0    10.1.1.4                 0             0 100 ?
Route Distinguisher: 2:2 (default for vrf cust2)
*> 0.0.0.0          0.0.0.0                  0         32768 ?
*> 192.168.111.0    0.0.0.0                  0         32768 ?
*> 192.168.123.0    10.1.1.3                 0             0 100 ?
*> 192.168.124.0    10.1.1.4                 0             0 100 ?
Device# show ip bgp vpnv4 all 192.168.113.1

BGP routing table entry for 1:1:192.168.113.0/24, version 48
Paths: (1 available, best #1, table cust1)
```

```
  Advertised to update-groups:
     3
  Refresh Epoch 1
  100
    10.1.1.3 from *10.1.1.3 (172.16.1.103)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:1:1
      mpls labels in/out 25/16
BGP routing table entry for 2:2:0.0.0.0/0, version 8
Paths: (1 available, best #1, table cust2)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.0.0.1)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:2:2
      mpls labels in/out 20/aggregate(cust2)
```

Additional References for Configuring MPLS over FlexVPN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 5586	*MPLS Generic Associated Channel*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring MPLS over FlexVPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 282: Feature Information for Configuring MPLS over FlexVPN

Feature Name	Releases	Feature Information
MPLS over FlexVPN		The following commands were introduced or modified: **clear ip nhrp, clear ipv6 nhrp, mpls nhrp, show dmvpn, show ip nhrp, show ipv6 nhrp.**

CHAPTER 214

Configuring IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices.

Information About IKEv2 Packet of Disconnect

Disconnect Request

The Packet of Disconnect (POD) is a RADIUS disconnect_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect a crypto session.

When the POD is Needed

The Packet of Disconnect is required in the following situations:

- Enforce reauthentication—As a network administrator, you might want to terminate a user on FlexVPN server to forcefully reauthenticate if a session is connected for a very long duration.
- Apply a new policy—As a network administrator, you may want to terminate an active crypto session and apply the new policy on the session when the client reconnects.
- Free resources—A session may need to be terminated to free resources and exit rekey.

IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature uses the RADIUS Packet of Disconnect (POD) feature to delete a crypto session. The crypto session is deleted to update VPN users to the new user or group policy on the AAA server.

1. AAA passes the attribute key-value pair list, provided by the RADIUS server, to IKEv2.

2. IKEv2 parses the list and locates the Audit-Session-ID, a Cisco AV pair, as a key and validates the pair value.
3. IKEv2 searches the session and deletes the specific session.
4. IKEv2 notifies AAA and AAA notifies the RADIUS server.
5. The session pertaining to the Audit-Session-ID is deleted.

Parameters in IKEv2 Packet of Disconnect

RFC 3576 specifies the following POD codes that are supported for IKEv2 Packet of Disconnect:

- 40 - Disconnect-Request
- 41 - Disconnect-ACK
- 42 - Disconnect-NAK

The Disconnect-ACK code indicates that a session existed for an audit-session-ID and that the session, pertaining to an audit-session-ID was terminated successfully. The Disconnect-NACK code indicates that there are no session corresponding to the audit-session-ID. No reply message is sent to the gateway.

How to Configure IKEv2 Packet of Disconnect

Configuring AAA on the FlexVPN Server

There is no IKEv2-specific configuration required on the FlexVPN server for the IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature. You only need to configure authentication, authorization, and accounting (AAA) on the FlexVPN server. For additional information on AAA configuration, see .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*hostname* | *ip-address*} [**server-key** *string* | **vrf** *vrf-id*]
6. **port** *number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa server radius dynamic-author** **Example:**	Sets up the local AAA server for the dynamic authorization service and enters dynamic authorization local server configuration mode. • In this mode, the RADIUS application commands are configured.
Step 5	**client** {*hostname* | *ip-address*} [**server-key** *string* | **vrf** *vrf-id*] **Example:** `Device(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco`	Configures the IP address or hostname of the AAA server client. • Use the **server-key** keyword and *string* argument to configure the server key at the client level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.
Step 6	**port** *number* **Example:** `Device(config-locsvr-da-radius)# port 1812`	Configures the UDP port.
Step 7	**end** **Example:** `Device(config-locsvr-da-radius)# end`	Exits dynamic authorization local server configuration mode and returns to privileged EXEC mode.

Configuration Examples for IKEv2 Packet of Disconnect

Example: Terminating an IKEv2 Session

The following is a sample output from the **show aaa sessions** command. This command must be executed to identify the IKEv2 session that needs to be terminated.

```
Device# show aaa sessions

Total sessions since last reload: 32
Session Id: 3
   Unique Id: 14
   User Name: *not available*
```

```
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
Session Id: 30
   Unique Id: 41
   User Name: pskuser2.g1.engdt.com
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
Session Id: 32
   Unique Id: 43
   User Name: pskuser4.g2.engdt.com
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
```

In the above output, ID 41 and 43 pertain to IKEv2 sessions. Optionally, you can run the **show aaa user** command to view detailed information about the session.

```
Device# show aaa user 41

Unique id 41 is currently in use.
  No data for type 0
  No data for type EXEC
  No data for type CONN
  NET: Username=(n/a)
    Session Id=0000001E Unique Id=00000029
    Start Sent=0 Stop Only=N
    stop_has_been_sent=N
    Method List=0
    Attribute list:
      7FBD9783CCF0 0 00000001 session-id(408) 4 30(1E)
      7FBD9783CD30 0 00000001 start_time(418) 4 Nov 04 2014 00:20:23
--------
  No data for type CMD
  No data for type SYSTEM
  No data for type VRRS
  No data for type RM CALL
  No data for type RM VPDN
  No data for type AUTH PROXY
  No data for type DOT1X
  No data for type CALL
  No data for type VPDN-TUNNEL
  No data for type VPDN-TUNNEL-LINK
  IPSEC-TUNNEL: Username=pskuser2.g1.engdt.com
    Session Id=0000001E Unique Id=00000029
    Start Sent=1 Stop Only=N
    stop_has_been_sent=N
    Method List=7FBDA6E05A68 : Name = accnt_prof
    Attribute list:
      7FBD9783CCF0 0 00000001 session-id(408) 4 30(1E)
      7FBD9783CD30 0 00000001 start_time(418) 4 Nov 04 2014 00:20:23
      7FBD9783CD70 0 00000082 formatted-clid(37) 13 192.168.202.2
     7FBD9783CDB0 0 0000008A audit-session-id(819) 37 L2L433010101ZO2L4C0A8CA02ZH119404ZP37

      7FBD9783CDF0 0 00000081 isakmp-phase1-id(737) 21 pskuser2.g1.engdt.com
      7FBD9783BF80 0 00000002 isakmp-initator-ip(738) 4 192.168.202.2
--------
  No data for type MCAST
  No data for type RESOURCE
  No data for type SSG
  No data for type IDENTITY
  No data for type ConnectedApps
```

```
Accounting:
  log=0x400018041
  Events recorded :
    CALL START
    ATTR REPLACE
    INTERIM START
    INTERIM STOP
    IPSEC TNL UP
  update method(s) :
    NONE
  update interval = 0
  Outstanding Stop Records : 0
  Dynamic attribute list:
    7FBD9783BF80 0 00000001 connect-progress(75) 4 No Progress
    7FBD9783BFC0 0 00000001 pre-session-time(334) 4 0(0)
    7FBD9783C000 0 00000001 elapsed_time(414) 4 341(155)
    7FBD9783C040 0 00000001 bytes_in(146) 4 0(0)
    7FBD9783C080 0 00000001 bytes_out(311) 4 0(0)
    7FBD9783CCF0 0 00000001 pre-bytes-in(330) 4 0(0)
    7FBD9783CD30 0 00000001 pre-bytes-out(331) 4 0(0)
    7FBD9783CD70 0 00000001 paks_in(147) 4 0(0)
    7FBD9783CDB0 0 00000001 paks_out(312) 4 0(0)
    7FBD9783CDF0 0 00000001 pre-paks-in(332) 4 0(0)
    7FBD9783BA20 0 00000001 pre-paks-out(333) 4 0(0)
Debg: No data available
Radi: No data available
Interface:
  TTY Num = -1
  Stop Received = 0
  Byte/Packet Counts till Call Start:
    Start Bytes In = 0             Start Bytes Out = 0
    Start Paks  In = 0             Start Paks  Out = 0
  Byte/Packet Counts till Service Up:
    Pre Bytes In = 0             Pre Bytes Out = 0
    Pre Paks  In = 0             Pre Paks  Out = 0
  Cumulative Byte/Packet Counts :
    Bytes In = 0             Bytes Out = 0
    Paks  In = 0             Paks  Out = 0
  StartTime = 00:20:23 IST Nov 4 2014
  AuthenTime = 00:20:23 IST Nov 4 2014
  Component = VPN IPSEC
Authen: service=NONE type=NONE method=NONE
Kerb: No data available
Meth: No data available
Preauth: No Preauth data.
General:
  Unique Id = 00000029
  Session Id = 0000001E
  Session Server Key = 1771D693
  Attribute List:
PerU: No data available
Service Profile: No Service Profile data.
Unkn: No data available
Unkn: No data available
```

Note the audit-session-id in the above output, which is L2L433010101ZO2L4C0A8CA02ZH119404ZP37. The following sample output is displayed on the FlexVPN server on starting an accounting session starts with a RADIUS server.

```
Nov  4 00:26:49.908 IST: RADIUS/ENCODE: Best Local IP-Address 192.168.202.1 for Radius-Server
 9.45.15.144
Nov  4 00:26:49.908 IST: RADIUS(0000002C): Send Accounting-Request to 9.45.15.144:1813 id
1646/231, len 288
```

```
Nov  4 00:26:49.908 IST: RADIUS:  authenticator 29 63 0C 79 C1 5E F2 0E - F3 CA 36 DD A3
55 C1 DE
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Session-Id     [44]  10  "00000021"
Nov  4 00:26:49.908 IST: RADIUS:  Calling-Station-Id  [31]  15  "192.168.202.2"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  64
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   58
"audit-session-id=L2L433010101ZO2L4C0A8CA02ZH11941194ZN3A"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  46
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   40
"isakmp-phase1-id=pskuser1.g1.engdt.com"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  40
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   34
"isakmp-initator-ip=192.168.202.2"
Nov  4 00:26:49.908 IST: RADIUS:  User-Name           [1]   23  "pskuser1.g1.engdt.com"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  36
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No Progress"
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Authentic      [45]  6   Local
[2]
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Status-Type    [40]  6   Start
[1]
Nov  4 00:26:49.908 IST: RADIUS:  NAS-IP-Address      [4]   6   192.168.202.1
Nov  4 00:26:49.908 IST: RADIUS:  home-hl-prefix      [151] 10  "D33648D8"
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Delay-Time     [41]  6   0
Nov  4 00:26:49.908 IST: RADIUS(0000002C): Sending a IPv4 Radius Packet
```

The following output is displayed on the FlexVPN server when disconnecting a session for a specific audit-session-id. The terminate session request is sent to the RADIUS server via a RADIUS client. In this example, the session for the audit-session-ID, which is L2L433010101ZO2L4C0A8CA02ZH119404ZP37 is terminated and, hence, not visible in the output.

```
Nov  4 00:32:29.004 IST: RADIUS: POD  received from id 216 9.45.15.144:50567, POD Request,
 len 84
Nov  4 00:32:29.004 IST: POD: 9.45.15.144 request queued
Nov  4 00:32:29.004 IST:  ++++++ POD Attribute List ++++++
Nov  4 00:32:29.004 IST: 7FBD9783D3A8 0 00000089 audit-session-id(819) 39
L2L433010101ZO2L4C0A8CA02ZH11941194ZN3B
Nov  4 00:32:29.004 IST:
Nov  4 00:32:29.004 IST: POD: Sending ACK from port 1812 to 9.45.15.144/50567

Nov  4 00:32:29.005 IST: IKEv2:(SESSION ID = 59,SA ID = 2):Check for existing active SA
Nov  4 00:32:29.006 IST: IKEv2:in_octets 0, out_octets 0
Nov  4 00:32:29.006 IST: IKEv2:in_packets 0, out_packets 0
Nov  4 00:32:29.006 IST: IKEv2:(SA ID = 2):[IKEv2 -> AAA] Accounting stop request sent
successfully
Nov  4 00:32:29.006 IST: IKEv2:(SESSION ID = 59,SA ID = 2):Delete all IKE SAs
Nov  4 00:32:29.010 IST: RADIUS/ENCODE(0000002D):Orig. component type = VPN IPSEC
Nov  4 00:32:29.010 IST: RADIUS(0000002D): Config NAS IP: 0.0.0.0
Nov  4 00:32:29.010 IST: RADIUS(0000002D): Config NAS IPv6: ::
Nov  4 00:32:29.010 IST: RADIUS(0000002D): sending
Nov  4 00:32:29.011 IST: RADIUS/ENCODE: Best Local IP-Address 192.168.202.1 for Radius-Server
 9.45.15.144
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Send Accounting-Request to 9.45.15.144:1813 id
1646/246, len 356
Nov  4 00:32:29.011 IST: RADIUS:  authenticator 52 88 5E CB 8B FA 1E C1 - CC EF 73 75 89
73 CA 95
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Session-Id     [44]  10  "00000022"
Nov  4 00:32:29.011 IST: RADIUS:  Calling-Station-Id  [31]  15  "192.168.202.2"
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  64
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   58
"audit-session-id=L2L433010101ZO2L4C0A8CA02ZH11941194ZN3B"
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  46
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   40
"isakmp-phase1-id=pskuser1.g1.engdt.com"
```

```
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  40
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   34
"isakmp-initator-ip=192.168.202.2"
Nov  4 00:32:29.011 IST: RADIUS:  User-Name           [1]   23  "pskuser1.g1.engdt.com"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Authentic      [45]  6   Local
[2]
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  36
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair      [1]   30  "connect-progress=No Progress"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Session-Time   [46]  6   56
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Input-Octets   [42]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Output-Octets  [43]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Input-Packets  [47]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Output-Packets [48]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Terminate-Cause[49]  6   none
[0]
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  32
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   26  "disc-cause-ext=No Reason"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Status-Type    [40]  6   Stop
[2]
Nov  4 00:32:29.011 IST: RADIUS:  NAS-IP-Address      [4]   6   192.168.202.1
Nov  4 00:32:29.011 IST: RADIUS:  home-hl-prefix      [151] 10  "E2F80C34"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Delay-Time     [41]  6   0
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Sending a IPv4 Radius Packet
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Started 5 sec timeout
```

The following output is displayed when there is no valid session for the specific audit-session-ID. This happens if there is no session pertaining to the specific audit-session-id when the session is terminated already. Note the NACK message that is sent back to the FlexVPN server

```
Nov  4 00:30:31.905 IST: RADIUS: POD  received from id 131 9.45.15.144:52986, POD Request,
 len 84
Nov  4 00:30:31.905 IST: POD: 9.45.15.144 request queued
Nov  4 00:30:31.905 IST:  ++++++ POD Attribute List ++++++
Nov  4 00:30:31.905 IST: 7FBD9783BA20 0 00000089 audit-session-id(819) 39
L2L433010101ZO2L4C0A8CA02ZH11941194ZN3A
Nov  4 00:30:31.905 IST:
Nov  4 00:30:31.906 IST: POD: 9.45.15.144 Unsupported attribute type 26 for component
Nov  4 00:30:31.906 IST: POD: 9.45.15.144 user  0.0.0.0i sessid 0x0 key 0x0 DROPPED
Nov  4 00:30:31.906 IST: POD: Added Reply Message: No Matching Session
Nov  4 00:30:31.906 IST: POD: Added NACK Error Cause: Invalid Request
Nov  4 00:30:31.906 IST: POD: Sending NAK from port 1812 to 9.45.15.144/52986
Nov  4 00:30:31.906 IST: RADIUS:  18  21  4E6F204D61746368696E672053657373696F6E
Nov  4 00:30:31.906 IST: RADIUS:  101 6   00000194
```

Additional References for IKEv2 Packet of Disconnect

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
RADIUS Packet of Disconnect	RADIUS Packet of Disconnect RADIUS Packet of Disconnect

Standards and RFCs

Standard/RFC	Title
RFC 3576	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*
RFC 5176	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IKEv2 Packet of Disconnect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 283: Feature Information for IKEv2 Packet of Disconnect

Feature Name	Releases	Feature Information
IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect		The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices. No commands were introduced by this feature.

CHAPTER 215

Configuring IKEv2 Change of Authorization Support

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session.

Prerequisites for IKEv2 Change of Authorization Support

- IKEv2 must be registered as a component, via a registry entry, on Cisco AAA component.

Restrictions for IKEv2 Change of Authorization Support

- This feature supports change of authorization (CoA) packets received from RADIUS-based AAA server only.

Information About IKEv2 Change of Authorization Support

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. When a policy changes for a user or user group in AAA, administrators can send the RADIUS CoA packets from the AAA server such as a Cisco Secure Access Control Server (ACS) to reinitialize authentication and apply the new policy.

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco software supports the RADIUS CoA request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Fore more information on RADIUS CoA, see *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS Release 15M&T* or *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS XE Release 3S*

Working of Change of Authorization on IKEv2

The FlexVPN - IKEv2 CoA for QoS and ACL feature allows to change attributes of an active IKEv2 crypto session to apply a new authorization attributes. The Cisco AAA component receives a Change of Authorization (CoA) packet from a AAA server and checks if the received CoA packet is meant for any of the components registered with it. If a component sees that the CoA packet is meant for itself, it processes it further. Based on the fields in the CoA packet, if the packet is relevant for a given component, such as IKEv2, the packet is consumed by that component. AAA will not forward the packet to the next component in the list.

In case of this feature, after IKEv2 receives a CoA packet, IKEv2 verifies the CoA packet for the Cisco (AV) pairs. IKEv2 identifies the session based on the audit-session-id which is already stored in the RADIUS server.

If the CoA packet contains attributes not supported by IKEv2, IKEv2 discards the packet and sends a CoA-NACK to AAA component.

Supported AV Pairs for IKEv2 Change of Authorization

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports the following Cisco AV pairs:

- ip:interface-config
- ip:sub-policy-In
- ip:sub-policy-Out
- ip:sub-qos-policy-in
- ip:sub-qos-policy-out
- ipsec:inacl
- ipsec:outacl

How to Configure IKEv2 Change of Authorization Support

Configuring Change of Authorization on the FlexVPN Server

There is no IKEv2-specific configuration required for this feature. on the FlexVPN server for the IKEv2 Change of Authorization (CoA) Support feature. You only need to configure the RADIUS Change of Authorization on the FlexVPN server. For more information on AAA configuration, see the RADIUS Change

of Authorization feature module in the *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS Release 15M&T*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string*
6. **port** *port-number*
7. **auth-type** {**any** | **all** | **session-key**}
8. **ignore session-key**
9. **ignore server-key**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Enters dynamic authorization local server configuration mode and specifies a RADIUS client from which a device accepts Change of Authorization (CoA) and disconnect requests. Configures the device as a AAA server to facilitate interaction with an external policy server.
Step 5	**client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string* **Example:** `Device(config-locsvr-da-radius)# client 10.0.0.1`	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 6	**port** *port-number* **Example:** `Device(config-locsvr-da-radius)# port 3799`	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients. **Note** The default port for packet of disconnect is 1700. Port 3799 is required to interoperate with ACS 5.1.

	Command or Action	Purpose
Step 7	**auth-type** {**any** | **all** | **session-key**} **Example:** `Device(config-locsvr-da-radius)# auth-type all`	Specifies the type of authorization that the device must use for RADIUS clients. The client must match the configured attributes for authorization.
Step 8	**ignore session-key** **Example:** `Device(config-locsvr-da-radius)# ignore session-key`	(Optional) Configures the device to ignore the session key.
Step 9	**ignore server-key** **Example:** `Device(config-locsvr-da-radius)# ignore server-key`	(Optional) Configures the device to ignore the server key.
Step 10	**exit** **Example:** `Device(config-locsvr-da-radius)# exit`	Returns to the global configuration mode.

Verifying IKEv2 Change of Authorization Support

Use the following show commands to view the success of change of authorization (CoA) on Cisco Devices.

SUMMARY STEPS

1. **enable**
2. **show platform hardware qfp active feature qos all output all**
3. **show platform hardware qfp active feature qos all input all**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show platform hardware qfp active feature qos all output all**

Example:

```
Device# show platform hardware qfp active feature qos all output all

Interface: Virtual-Access1, QFP if_h: 14, Num Targets: 1
  Target: Out, Num UIDBs: 1
    UIDB #: 0
    Hierarchy level: 0, Num matching iftgts: 1
    Policy name: aaa-out-policy, Policy id: 9679472
```

```
    Parent Class Idx: 0, Parent Class ID: 0
      IF Tgt#: 0, ifh: 14, member_ifh: 0, link_idx: 0
        PSQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593, Match index: 0
            Class name: class-default, Policy name: aaa-out-policy
              psqd[0-3]: 0x00000000 0x00000000 0x00000001 0x00000000
        ISQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              isqd[0-3]:  0x88e78ec0 0x00000000 0x00000000 0x00000000
              (cache) isqd[0-3]: 0x88e78ec0 0x00000000 0x00000000 0x00000000
        Police specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              Policer id: 0x20000002
              hw_policer[0-3]:       0x4000047e 0x00163ac8 0x00000000 0x00000000
              cache hw_policer[0-3]: 0x4000047e 0x00163ac8 0x00000000 0x00000000
              conform stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
              exceed stats  (paks/octets): 0x0000000000000000, : 0x0000000000000000
              violate stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
              police_info:           0x00000000
              cache police_info:     0x00000000
        Queue specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              No queue configured
        Schedule specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              No schedule info (no queue configured)
```

Displays platform-specific information if CoA was successful.

Step 3 **show platform hardware qfp active feature qos all input all**

Example:

```
Device#  show platform hardware qfp active feature qos all input all

Interface: Virtual-Access1, QFP if_h: 14, Num Targets: 1
  Target: In, Num UIDBs: 1
    UIDB #: 0
    Hierarchy level: 0, Num matching iftgts: 1
    Policy name: aaa-in-policy, Policy id: 980784
    Parent Class Idx: 0, Parent Class ID: 0
      IF Tgt#: 0, ifh: 14, member_ifh: 0, link_idx: 0
        PSQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593, Match index: 0
            Class name: class-default, Policy name: aaa-in-policy
              psqd[0-3]: 0x00000000 0x00000000 0x00000001 0x00000000
        ISQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-in-policy
              isqd[0-3]:  0x88d49748 0x00000001 0x00000000 0x00000000
              (cache) isqd[0-3]: 0x88d49748 0x00000001 0x00000000 0x00000000
        Police specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
```

```
    Class name: class-default, Policy name: aaa-in-policy
      Policer id: 0x20000003
      hw_policer[0-3]:       0x10000140 0x00113a29 0x00000000 0x00000000
      cache hw_policer[0-3]: 0x10000140 0x00113a29 0x00000000 0x00000000
      conform stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
      exceed stats  (paks/octets): 0x0000000000000000, : 0x0000000000000000
      violate stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
      police_info:           0x00000000
      cache police_info:     0x00000000
Queue specifics:
  Target Index: 0, Num Classes: 1
    Class index: 0, Class object id: 1593
    Class name: class-default, Policy name: aaa-in-policy
      No queue configured
Schedule specifics:
  Target Index: 0, Num Classes: 1
    Class index: 0, Class object id: 1593
    Class name: class-default, Policy name: aaa-in-policy
      No schedule info (no queue configured)
```

Displays the feature status.

Configuration Examples for IKEv2 Change of Authorization Support

Example: Triggering a Change of Authorization

The following sample output is displayed when an administrator triggers a change of authorization (CoA). The session is identified based on the audit-session-id, a dynamic string, which is an encoded form of 6 tuple information of a session with peer.

IKEv2 receives a change of authorization (CoA) packet from a RADIUS server. The session is identified based on audit-session-id.

```
*Oct  6 23:38:55.250: RADIUS: COA  received from id 125 10.106.210.176:58712, CoA Request,
 len 257
*Oct  6 23:38:55.251: COA: 10.106.210.176 request queued
*Oct  6 23:38:55.251: RADIUS:  authenticator BD 97 5E BA B2 EB C1 C5 - 1A 14 51 3D C2 C8
66 3F
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  62
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   56
"audit-session-id=L2L44D010102ZO2L44D010101ZI1F401F4ZO2"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  52
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   46
"ip:interface-config=service-policy input pol"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  35
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   29  "ip:sub-qos-policy-out=2M-IN"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  36
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   30  "ip:sub-qos-policy-in=aaa-pol"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  52
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   46
"ip:interface-config=service-policy output 2M"
*Oct  6 23:38:55.251: COA: Message Authenticator missing or failed decode

*Oct  6 23:38:55.251:  ++++++ CoA Attribute List ++++++
```

```
*Oct  6 23:38:55.251: 421C9694 0 00000089 audit-session-id(819) 37
L2L44D010102ZO2L44D010101ZI1F401F4ZO2
*Oct  6 23:38:55.251: 421C9584 0 00000081 interface-config(222) 24 service-policy input pol
*Oct  6 23:38:55.251: 421C95B8 0 00000081 sub-qos-policy-out(423) 5 2M-IN
*Oct  6 23:38:55.251: 421C95EC 0 00000081 sub-qos-policy-in(421) 7 aaa-pol
*Oct  6 23:38:55.251: 421C9620 0 00000081 interface-config(222) 24 service-policy output
2M
*Oct  6 23:38:55.251:
*Oct  6 23:38:55.251: COA: Added NACK Error Cause: Success
```

Additional References for IKEv2 Change of Authorization Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IKEv2 Change of Authorization Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 284: Feature Information for IKEv2 Change of Authorization Support

Feature Name	Releases	Feature Information
FlexVPN - IKEv2 CoA for QoS and ACL		The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session. No commands were modified or updated by this feature.

CHAPTER 216

Configuring Aggregate Authentication

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server.

Prerequisites for Configuring Aggregate Authentication

- You must enable the BypassDownloader function in the AnyConnectLocalPolicy file by setting the <BypassDownloader> value to true. If your device does not support SSL, the BypassDownloader function will not work.You must disable the function manually by setting the <BypassDownloader> value to false, else the connection will fail.

Information for Configuring Aggregate Authentication

Cisco AnyConnect and FlexVPN

To establish a VPN connection, the VPN client must obtain user credentials using authentication methods such as, extensible authentication protocol (EAP), Extended Authentication (XAUTH), etc. and forward the user credentials to a hub, which contacts an access control server. The access control server sends an external database or active directory (AD) to validate the credentials.

FlexVPN server (as a hub) works with Cisco Secure Access Control Server to validate user credentials to establish VPN connections. However, Cisco AnyConnect uses EAP to obtain user credentials and does not support XAUTH. On the other hand, Cisco Secure Access Control Server does not support EAP-MD5 with external database (in this case AD). This leads to a scenario where either Cisco Secure Access Control Server must support EAP-MD5 or FlexVPN must authenticate the information from Cisco AnyConnect separately and connect separately with Cisco Secure Access Control Server. FlexVPN can use the Aggregate

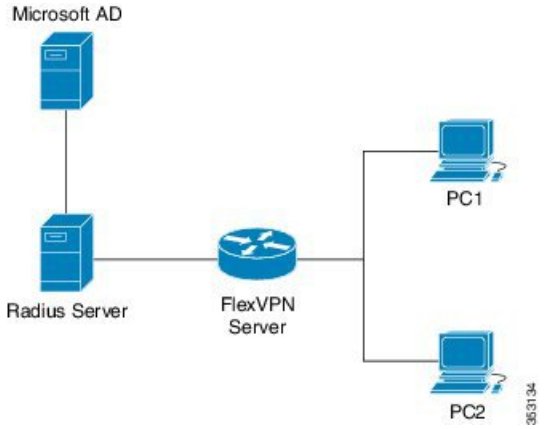
Authentication method to authentication information from Cisco AnyConnect. Implementing aggregate authentication method on FlexVPN server would provide a window to add more feature support on Cisco IOS software.

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet using Cisco AnyConnect and FlexVPN server. This is a server-specific feature and works with Cisco AnyConnect.

How Aggregate Authentication Works

Internet Key Exchange Version 2 supports Cisco AnyConnect that uses the proprietary AnyConnect EAP authentication method by implementing basic aggregate authentication where authentication is performed via authentication, authorization, and accounting (AAA) using the remote RADIUS server. The following is an example of a network topology explains aggregate authentication implementation on Cisco IOS software.

Figure 111: FlexVPN Server Connected to RADIUS Server



In this diagram:

- Cisco Secure Access Control Server acts as a RADIUS server for authorization.
- The credentials are stored in Microsoft Active Directory, which acts as the active directory for authentication.

Note Microsoft Active Directory is referred for example purpose only. It does not matter where the credentials are stored.

- Cisco device acts as FlexVPN server.
- Windows 7 PC acts as Cisco AnyConnect client.

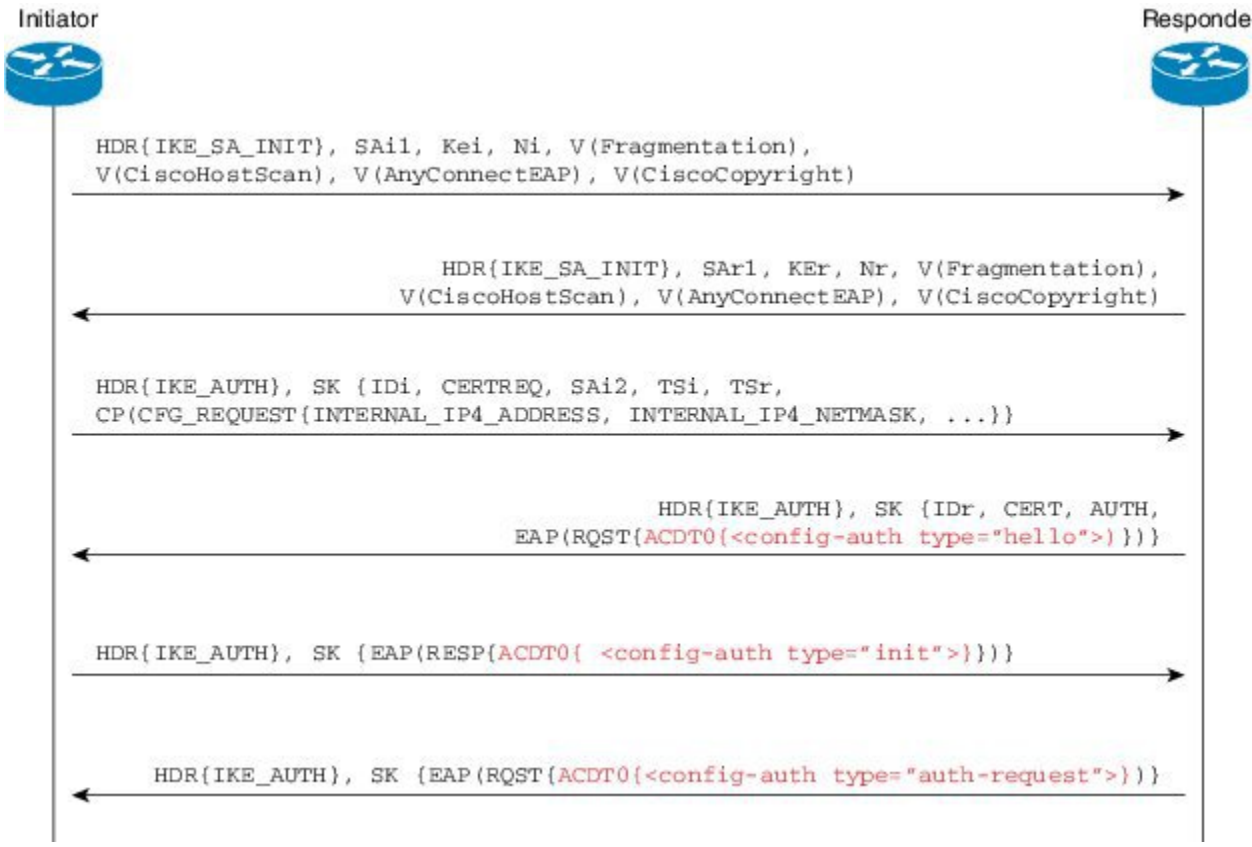
1. To initiate a VPN connection, Cisco AnyConnect client verifies a FlexVPN server using certifications.
2. After verifying the certificates, Cisco AnyConnect client sends Cisco AnyConnect EAP loaded message to FlexVPN server.

3. When FlexVPN server receives Cisco AnyConnect EAP loaded message from Cisco AnyConnect, FlexVPN server downloads the message and strips the message of EAP.
4. FlexVPN establishes a connection with RADIUS server for authorization and Microsoft Active Directory (AD) for authentication, and forwards the stripped message to verify the credentials provided by Cisco AnyConnect client.
5. When the credentials are verified and approved by RADIUS server and Microsoft Active Directory (AD), an appropriate reply is sent to FlexVPN server, which in turn replies to Cisco AnyConnect and a VPN connection is established.

IKE Exchanges Using Cisco AnyConnect EAP

Authentication in IKE using AnyConnect EAP is a variation of the standards EAP model as described in RFC 3748. When using AnyConnect EAP the public configuration or authentication XML is transported via EAP payloads. The following figure illustrates the typical message flow used by Cisco AnyConnect .

Figure 112: IKE Exchanges using AnyConnect EAP



1. Cisco AnyConnect client initiates IKE connection to FlexVPN server. The client sends vendor ID payloads to indicate support for Cisco AnyConnect EAP in addition to the typical IKE payloads. The client identifies itself as a Cisco product by including the Cisco copyright vendor ID.
2. The server gateway sends vendor ID payloads to indicate fragmentation and AnyConnect EAP support and identifies itself as a Cisco product by including the Cisco copyright vendor ID.

3. The configuration payload requests the tunnel configuration. The client indicates its desire to use Cisco AnyConnect EAP authentication by omitting the AUTH Payload from this message.
4. The Aggregate Authentication and Configuration protocol is carried over EAP
5. FlexVPN server sends a EAP success message.
6. Cisco AnyConnect client sends the AUTH payload.
7. FlexVPN server sends the AUTH payload and the tunnel configuration attributes that Cisco AnyConnect client requested.

Dual-Factor Authentication Support with IKEv2

The aggregate authentication implementation on Cisco IOS software can be extended for dual-factor authentication. Double authentication can be done by introducing new AnyConnect EAP exchange during Aggregate Authentication which exchange and validate the device certificate information. This mechanism of authenticating 'device' as well as 'user' is called 'Double Authentication'.

Note AnyConnect EAP is AnyConnect client specific authentication method and does not apply to any other client.

How to Configure Aggregate Authentication

Configuring the FlexVPN Server for Aggregate Authentication

Perform this task to configure aggregate authentication on the FlexVPN server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **aaa accounting anyconnect-eap** *list-name*
5. **match identity remote key-id** *opaque-string*
6. **authentication remote anyconnect-eap aggregate** [**cert-request**]
7. **authentication local rsa-sig**
8. **pki trustpoint** *trustpoint-label*
9. **aaa authentication anyconnect-eap** *list-name*
10. **aaa authorization group anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name*
11. **aaa authorization user anyconnect-eap cached**
12. **aaa authorization user anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name*
13. **end**
14. **show crypto ikev2 session detailed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile name and enters IKEv2 profile configuration mode.
Step 4	**aaa accounting anyconnect-eap** *list-name* **Example:** `Device(config-ikev2-profile)# aaa accounting anyconnect-eap list1`	Enables authentication, authorization, and accounting (AAA) accounting method lists when the IKEv2 remote authentication method is AnyConnect EAP.
Step 5	**match identity remote key-id** *opaque-string* **Example:** `Device(config-ikev2-profile)# match identity remote key-id aggauth_user3@abc.com`	Matches a profile based on the identity of the type remote key ID.
Step 6	**authentication remote anyconnect-eap aggregate** [**cert-request**] **Example:** `Device(config-ikev2-profile)# authentication remote anyconnect-eap aggregate cert-request`	Specifies aggregate authentication for Cisco AnyConnect EAP. • **cert-request** - requests certificate from Cisco AnyConnect client for double authentication.
Step 7	**authentication local rsa-sig** **Example:** `Device(config-ikev2-profile)# authentication local rsa-sig`	Specifies Rivest, Shamir, and Adelman (RSA) signature as the local authentication method.
Step 8	**pki trustpoint** *trustpoint-label* **Example:** `Device(config-ikev2-profile)# pki trustpoint CA1`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method.
Step 9	**aaa authentication anyconnect-eap** *list-name* **Example:** `Device(config-ikev2-profile)# aaa authentication anyconnect-eap list1`	Specifies authentication, authorization, and accounting (AAA) authentication list for Cisco AnyConnect EAP authentication. • **anyconnect-eap**—Specifies AAA AnyConnect EAP authentication. • *list-name*—The AAA authentication list name.

	Command or Action	Purpose
Step 10	**aaa authorization group anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name* **Example:** `Device(config-ikev2-profile)# aaa authorization group anyconnect-eap list list1 name-mangler mangler1`	Specifies the AAA authorization for each group policy when the remote authentication method is AnyConnect EAP and derives the name mangler.
Step 11	**aaa authorization user anyconnect-eap cached** **Example:** `Device(config-ikev2-profile)# aaa authorization user anyconnect-eap cached`	Specifies the AAA authorization for each user policy when the remote authentication method is AnyConnect EAP and uses cached attributes from the AnyConnect EAP authentication.
Step 12	**aaa authorization user anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name* **Example:** `Device(config-ikev2-profile)# aaa authorization user anyconnect-eap list list1 name-mangler mangler1`	Specifies the AAA method list for the remote authentication method and derives the name mangler.
Step 13	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.
Step 14	**show crypto ikev2 session detailed** **Example:** `Device# show crypto ikev2 session detailed`	Displays the status of active Internet Key Exchange Version 2 (IKEv2) sessions.

Configuration Examples for Aggregate Authentication

Example: Configuring Aggregate Authentication

The following example shows how to configure aggregate authentication on the FlexVPN server to enable the establishment of a secure tunnel between Cisco AnyConnect Client and FlexVPN server.

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 profile profile1
Device(config-ikev2-profile)# aaa accounting anyconnect-eap list1
Device(config-ikev2-profile)# match identity remote key-id aggauth_user1@example.com
Device(config-ikev2-profile)# authentication remote anyconnect-eap aggregate cert-request
Device(config-ikev2-profile)# authentication local rsa-sig
Device(config-ikev2-profile)# pki trustpoint CA1
Device(config-ikev2-profile)# aaa authentication anyconnect-eap list1
Device(config-ikev2-profile)# aaa authorization group anyconnect-eap list list1 name-mangler
 mangler1
Device(config-ikev2-profile)# aaa authorization user anyconnect-eap cached
Device(config-ikev2-profile)# aaa authorization user anyconnect-eap list list1 name-mangler
```

```
 mangler1
Device(config-ikev2-profile)# end
```

Additional References for Configuring Aggregate Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Aggregate Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 285: Feature Information for Configuring Aggregate Authentication

Feature Name	Releases	Feature Information
Dual-Factor Authentication support with IKEv2		Dual-Factor Authentication support with IKEv2 supports certificate request from Cisco AnyConnect client for double authentication. The following command was modified: **authentication (IKEv2 profile)**.
FlexVPN RA - Aggregate Auth Support for AnyConnect		The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server. The following commands were introduced or modified: **aaa accounting (IKEv2 profile)**, **aaa authentication (IKEv2 profile)**, **aaa authorization (IKEv2 profile)**, **authentication (IKEv2 profile)**, **show crypto ikev2 profile**, **show crypto ikev2 session**.

CHAPTER 217

Appendix: FlexVPN RADIUS Attributes

This chapter describes the RADIUS attributes supported by FlexVPN server.

• FlexVPN RADIUS Attributes, on page 2829

FlexVPN RADIUS Attributes

The following are the RADIUS attributes categories used by FlexVPN Server:

- Inbound and bidirectional IETF RADIUS attributes
- Outbound Local
- Outbound Remote

Note For inbound attributes sent by the FlexVPN server to RADIUS that are not listed below, the value is set by the AAA system.

Attribute	User-Name
Type	IETF
Format	String
Attribute ID	1
Description	This attribute is sent by the FlexVPN server to Radius and is derived as follows: • AAA based preshared keys—Peer IKEv2 identity • EAP authentication—Peer EAP identity • User or group authorization—Output of the name mangler or the string specified in the IKEv2 profile authorization commands • Accounting—Peer EAP identity or IKEv2 identity This attribute may also be received from Radius in Access-Accept after successful EAP authentication and specifies the authenticated peer EAP identity.

Attribute	User-Password
Type	IETF
Format	String
Attribute ID	2
Description	This attribute is sent by the FlexVPN server to RADIUS and is derived as follows: • AAA based preshared keys—“cisco” • User/group authorization—“cisco”

Attribute	Calling-Station-ID
Type	IETF
Format	String
Attribute ID	31
Description	This attribute is sent by FlexVPN server to RADIUS and is derived as follows: • AAA based pre-shared keys—IKEv2 initiator address • EAP authentication—IKEv2 initiator address • User/group authorization—IKEv2 initiator address

Attribute	Service-Type
Type	IETF
Format	String
Attribute ID	6
Description	This attribute is used by FlexVPN server for EAP authentication and the value of this attribute is set to ‘Login’.

Attribute	EAP-Message
Type	IETF
Format	String
Attribute ID	79
Description	This attribute is used by FlexVPN server for EAP authentication to relay EAP packets between EAP server and the Remote Access Client.

Attribute	Message-Authenticator

Type	IETF
Format	String
Attribute ID	80
Description	This attribute is sent by FlexVPN server for EAP authentication. The value for this attribute is set by AAA subsystem.

Attribute	Framed-Pool
Type	IETF
Format	String
Attribute ID	88
Local config	pool name
Radius config	Framed-Pool=*pool-name*
Description	Specifies the name of IPv4 address pool that is used by FlexVPN server to allocate the IPv4 address to assign to the client. The allocated address is pushed to client via IKEv2 standard config attribute INTERNAL_IP4_ADDRESS.

Attribute	ipsec:group-dhcp-server
Type	Cisco AV Pair
Format	String
Local config	dhcp server {*ipddr* | *host*}
Radius config	cisco-avpair="ipsec: group-dhcp-server=*ipaddr*"
Description	Specifies the IPv4 DHCP server that is used by FlexVPN server to lease IPv4 address to assign to the client. The leased address is pushed to client via IKEv2 standard config attribute INTERNAL_IP4_ADDRESS.

Attribute	ipsec:dhcp-giaddr
Type	Cisco AV Pair
Format	IPaddr
Local config	dhcp giaddr ipaddr
Radius config	cisco-avpair="psec: dhcp-giaddr=*ipaddr*"
Description	Specifies the IPv4 DHCP gateway IP address that is used by FlexVPN server to contact the DCHP server.

Attribute	ipsec:dhcp-timeout

Type	Cisco AV Pair
Format	Integer
Local config	dhcp timeout *seconds*
Radius config	cisco-avpair="ipsec:dhcp-timeout=*seconds*"
Description	Specifies the time to wait for response from IPv4 DHCP server that is used by FlexVPN server to timeout response from the DHCP server.

Attribute	ipsec:ipv6-addr-pool
Type	Cisco AV Pair
Format	String
Local config	ipv6 *pool name*
Radius config	cisco-avpair="ipsec:ipv6-addr-pool=*pool-name*"
Description	Specifies the name of IPv6 address pool used by FlexVPN server to allocate the IPv6 address to assign to the client. The allocated address is pushed to the client via IKEv2 standard config attribute INTERNAL_IP6_ADDRESS.

Attribute	ipsec:route-set=prefix
Type	Cisco AV Pair
Format	String
Local config	N/A
Radius config	cisco-avpair="ipsec:route-set=prefix *prefix*/*length*"
Example	ipsec:route-set=prefix 192.168.1.0/24
Description	Specifies a subnet protected by FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. **Note** This AV pair was introduced in Cisco IOS Release 15.2(2)T.

Attribute	ipsec:route-set=interface
Type	Cisco AV Pair
Format	String
Local config	route set interface
Radius config	cisco-avpair="ipsec:route-set=interface"

Description	This attribute is used locally and enables sending of VPN interface IP address to the peer via IKEv2 standard config attribute INTERNAL_IP4_SUBNET. This allows running routing protocols such as BGP over VPN. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the "ipsec:route-set-interface" AV pair.

Attribute	ipsec:route-accept
Type	Cisco AV Pair
Format	String
Local config	route accept any [tag *tag-id*] [distance *distance*]
Radius config	cisco-avpair="ipsec:route-accept=any [tag:*tag*] [distance:*distance*]"
Example	ipsec:route-accept=any tag=100
Description	This attribute is used locally and specifies the filter for the subnets received from the peer via IKEv2 standard config attribute INTERNAL_IP4_SUBNET. The attribute also specifies the tag and distance for the routes added by IKEv2 for the filtered subnets. **Note** In Cisco IOS Release 15.2(2)T, the AV pair "ipsec:route-accept=any" replaced "ipsec:route-accept=accept acl:any" and the AV pair "ipsec:route-accept=none" replaced "ipsec:route-accept=deny".

Attribute	ipsec:ipsec-flow-limit
Type	Cisco AV Pair
Format	Integer
Local config	ipsec flow-limit *limit*
Radius config	cisco-avpair="ipsec:ipsec-flow-limit=*limit*"
Description	This attribute is used by FlexVPN server and specifies the maximum number of IPsec SAs that an IPSec dVTI session can have. There is no limit by default. This parameter is similar to the **crypto ipsec profile** and **set security-policy limit** commands.

Attribute	ip:interface-config
Type	Cisco AV Pair
Format	String
Local config	aaa attribute list *list* attribute type interface-config *string*
Radius config	cisco-avpair="ip:interface-config=interface cmd string"
Example	ip:interface-config=ip vrf forwarding red

Description	This attribute is used locally and specifies an interface configuration mode command string that is applied on the virtual access interface for the session. For local configuration, the IKEv2 authorization policy points to an AAA attribute list that must have interface-config attribute.

Attribute	Tunnel-Type
Type	IETF
Format	Integer
Attribute ID	64
Radius config	Tunnel-Type=type
Description	This attribute specifies the tunnel type (ESP, AH, GRE, etc.) and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	Tunnel-Medium-Type
Type	IETF
Format	Integer
Attribute ID	65,
Radius config	Tunnel-Medium-Type=type
Description	This attribute specifies the tunnel transport type (IPv4, IPv6, etc.) and is received when FlexVPN server fetches preshared key for the session from the RADIUS server.

Attribute	Tunnel-Password
Type	IETF
Format	String
Attribute ID	69
Radius config	Tunnel-Password=string
Description	This attribute specifies the symmetric preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	ipsec:ikev2-password-local
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:ikev2-password-local=*string*"
Description	This attribute specifies the local preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	ipsec:ikev2-password-remote
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:ikev2-password-remote=*string*"
Description	This attribute specifies the remote preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	Framed-IP-Address
Type	IETF
Format	IPaddr
Attribute ID	8
Radius config	Framed-IP-Address=*ipaddr*
Description	Specifies IPv4 address assigned to the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_ADDRESS.

Attribute	Framed-IP-Netmask
Type	IETF
Format	IPaddr
Attribute ID	9
Local config	netmask *mask*
Radius config	Framed-IP-Netmask=*mask*
Description	Specifies the subnet mask of the IPv4 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP4_NETMASK.

Attribute	ipsec:dns-servers
Type	Cisco AV Pair
Format	String
Local config	dns *primary* [*secondary*]
Radius config	cisco-avpair="ipsec:dns-servers=*primary secondary*"
Description	Specifies the primary and secondary IPv4 DNS servers for the client. This is pushed to the client via IKEv2 standard config attribute INTERNAL_IP4_DNS.

Attribute	ipsec:wins-servers

Type	Cisco AV Pair
Format	String
Local config	wins *primary* [*secondary*]
Radius config	cisco-avpair="ipsec:wins-servers=*primary secondary*"
Description	Specifies the primary and secondary IPv4 WINS servers for the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_NBNS.

Attribute	ipsec:route-set=access-list
Type	Cisco AV Pair
Format	String
Local config	route set access-list {*acl-name* | *acl-number*}
Radius config	cisco-avpair="ipsec:route-set=access-list {*acl-name* | *acl-number*}"
Description	Specifies the IPv4 subnets protected by FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the "ipsec:inacl" AV pair.

Attribute	ipsec:addrv6
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:addrv6=*ipv6-addr*"
Description	Specifies the IPv6 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP6_ADDRESS in the first 16 bytes.

Attribute	ipsec:prefix-len
Type	Cisco AV Pair
Format	Integer
Local config	N/A
Radius config	cisco-avpair="ipsec:prefix-len=*value*"
Example	ipsec:prefix-len=24
Description	Specifies the prefix length of the IPv6 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP6_ADDRESS in the last (17^{th}) byte.

Attribute	ipsec:ipv6-dns-servers-addr

Type	Cisco AV Pair
Format	String
Local config	ipv6 dns *primary* [*secondary*]
Radius config	cisco-avpair="ipsec: ipv6-dns-servers-addr=ipaddr1 *ipaddr2"
Description	Specifies the primary and secondary IPv6 DNS servers for the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP6_DNS.

Attribute	ipsec:route-set=access-list ipv6
Type	Cisco AV Pair
Format	String
Local config	route set access-list ipv6 acl-name
Radius config	cisco-avpair="ipsec:route-set=access-list ipv6 *acl-name*"
Description	Specifies IPv6 subnets protected by the FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP6_SUBNET. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the " ipsec:ipv6-subnet-acl" AV pair.

Attribute	ipsec:banner
Type	Cisco AV Pair
Format	String
Local config	banner *text*
Radius config	cisco-avpair="ipsec:banner=*text*"
Description	Specifies the banner text. This is pushed to the client via Cisco Unity attribute MODECFG_BANNER.

Attribute	ipsec:default-domain
Type	Cisco AV Pair
Format	String
Local config	def-domain *name*
Radius config	cisco-avpair="ipsec:default-domain=*name*"
Description	Specifies the default domain. This is pushed to the client via Cisco Unity attribute MODECFG_DEFDOMAIN.

Attribute	ipsec:split-dns

Type	Cisco AV Pair
Format	String
Local config	split-dns name
Radius config	cisco-avpair="ipsec:split-dns=name"
Description	Specifies the split DNS name. This is pushed to the client via Cisco Unity attribute MODECFG_SPLITDNS_NAME. You can configure up to 10 split DNS names.

Attribute	ipsec:ipsec-backup-gateway
Type	Cisco AV Pair
Format	String
Local config	backup-gateway *name*
Radius config	cisco-avpair="ipsec:ipsec-backup-gateway=*name*"
Description	Specifies the backup gateway. This is pushed to the client via Cisco Unity attribute MODECFG_BACKUPSERVERS. You can configure up to 10 backup gateways.

Attribute	ipsec:pfs
Type	Cisco AV Pair
Format	Integer
Local config	pfs
Radius config	cisco-avpair="ipsec:pfs=*value*"
Description	Specifies IPsec PFS (Perfect Forward Secrecy) enable/disable. This is pushed to the client via Cisco Unity attribute MODECFG_PFS. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:include-local-lan
Type	Cisco AV Pair
Format	Integer
Local config	include-local-lan
Radius config	cisco-avpair="ipsec:include-local-lan=*value*"
Description	Enables or disables include local LAN. This is pushed to the client via Cisco Unity attribute MODECFG_INCLUDE_LOCAL_LAN. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:smartcard-removal-disconnect
Type	Cisco AV Pair

Format	Integer
Local config	smartcard-removal-disconnect
Radius config	cisco-avpair="ipsec:smartcard-removal-disconnect =*value*"
Description	Enables or disables smartcard removal disconnect. This is pushed to the client via Cisco Unity attribute MODECFG_SMARTCARD_REMOVAL_DISCONNECT. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:configuration-url
Type	Cisco AV Pair
Format	String
Local config	configuration url *url*
Radius config	cisco-avpair="ipsec:configuration-url=*url*"
Description	Specifies the URL for configuration download. This is pushed to the client via Cisco FlexVPN attribute MODECFG_CONFIG_URL.

Attribute	ipsec:configuration-version
Type	Cisco AV Pair
Format	Integer
Local config	configuration version *version*
Radius config	cisco-avpair="ipsec:configuration-version=*version*"
Description	Specifies the version of the configuration to download. This is pushed to the client via Cisco FlexVPN attribute MODECFG_CONFIG_VERSION.

Attribute	Route-set remote
Type	Cisco AV Pair
Format	String
Local config	route set remote {ipv4 ip-address mask | ipv6 ip-address/mask}
Radius config	cisco-avpair="ipsec:route-set=remote {ipv4 network subnet_mask | ipv6 network/subnet_mask}"
Description	Specifies a subnet protected by FlexVPN server. This is pushed to the client through IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. While route-set prefix is working with a subnet mask represented in decimal fashion [e.g. /24], route-set remote requires the standard subnet mask representation. [e.g. 255.255.255.0] **Note** This AV pair was introduced in Cisco IOS Release 3.10.0S

Attribute	Route-set local
Type	Cisco AV Pair
Format	String
Local config	route set local {ipv4 ip-address mask | ipv6 ip-address/mask}
Radius config	cisco-avpair="ipsec:route-set=local {ipv4 network subnet_mask | ipv6 network/subnet_mask}”
Description	This attribute is useful in an extranet scenario where you do not necessary trust the routing information that you receive from the remote device. In other words, remote routes can be denied and route addition can be locally controlled by using this AV pair. **Note** This AV pair was introduced in Cisco IOS Release 3.10.0S.

CHAPTER 218

Appendix: IKEv2 and Legacy VPNs

This module provides examples on how to configure IKEv2 on crypto map based configurations.

Note Crypto maps are considered a legacy configuration construct. It is recommended that you migrate existing crypto map based setups to use tunnel protection and virtual interfaces.

Example: Configuring Crypto-Map-Based IKEv2 Peers Using Preshared Key Authentication Method

The following example shows how to configure crypto-map-based IKEv2 peers using the preshared key authentication method between a static crypto-map IKEv2 initiator and a dynamic crypto-map IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.231 255.255.255.224
  pre-shared-key abc
 !
!
!
```

```
crypto ikev2 profile prof
 match fvrf any
 match identity remote fqdn dmap-responder
 identity local fqdn smap-initiator
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.227 255.255.255.224
 crypto map cmap
!
ip route 209.165.200.229 255.255.255.224 209.165.200.225
!
ip access-list extended ikev2list
 permit ip any any
!
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.228
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote fqdn smap-initiator
 identity local fqdn dmap-responder
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
 ivrf global
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto dynamic-map dmap 1
 set transform-set trans
 set reverse-route tag 222
 set ikev2-profile prof
```

```
 match address ikev2list
!
crypto map cmap 1 ipsec-isakmp dynamic dmap
!
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
 crypto map cmap
!
ip route 209.165.200.233 255.255.255.224 209.165.200.228
!
ip access-list extended ikev2list
 permit ip any any
!
```

To initiate the connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

To display the session details, enter the following **show** commands:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 209.165.200.225 port 500
  IKEv2 SA: local 209.165.200.228/500 remote 209.165.200.231/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detail
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   209.165.200.228/500  209.165.200.231/500           (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK

      Life/Active Time: 86400/21 sec
      CE id: 1002, Session-id: 2
      Status Description: Negotiation done
      Local spi: 687752902752A6FD       Remote spi: C9DCCFC65493D14F
      Local id: smap-initiator
      Remote id: dmap-responder
      Local req msg id:  2              Remote req msg id:  0
      Local next msg id: 2              Remote next msg id: 0
      Local req queued:  2              Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring Crypto Map-Based IKEv2 Peers Using Certification Authentication Method

The following example shows how to configure crypto-map-based IKEv2 peers using the certificate authentication method between a static crypto-map IKEv2 initiator, a dynamic crypto-map IKEv2 responder, and a CA server. The initiator configuration is as follows:

```
crypto pki trustpoint ca-server
 enrollment url http://10.1.1.3:80
 revocation-check none
!
crypto pki certificate map cmap-1 1
 subject-name eq hostname = responder
!
!
crypto pki certificate chain ca-server
 certificate 02
  308201AF 30820118 A0030201 02020102 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32353132 355A170D 31313033 31303132 35313235 5A301A31 18301606 092A8648
  86F70D01 09021609 494E4954 4941544F 52305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100A47E 8C58BA89 8CCDC5A4 5A63BD29 C331A2A5 393F4616
  6B43FD2E 5ED4C81A 913E3B13 33A9B2DC CFC30391 24BB0DC8 B28FD6F1 C008D101
  34C10062 30F88CF7 9D630203 010001A3 4F304D30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 144871D9 002C66DF D85FACB8 45D1D25F EA357455
  91301D06 03551D0E 04160414 E77C74E7 183AB530 83DC531B 1DE3DA1D 914A925D
  300D0609 2A864886 F70D0101 04050003 81810042 21934B77 7E485E6F EE717D75
  6407B361 45190CEF E1A29CF2 6FA29E9A 5ECC1CEE B273533D 1453F6CE 1FDDA747
  7E701B4B 2A2AE53F D67C2345 952325BA 30950435 0706C5EE A7A8B414 CFEEB7A2
  9CD46F8F 3F663268 A20C4CCF E75D61EF 03FBA85D EDD6B26E 63653F09 F97DAFA6
  6C76E44E C9CA3FDC 6CD85D30 169A1D9E 4E870B
        quit
 certificate ca 01
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32343933 385A170D 31333033 30393132 34393338 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 DA4ECE09 B998F670 598F32C1 7E9FA920 1D217AC4 293B842E
  7563CE11 B2F0F822 23077930 636C8293 00F6CFDD F6C9B0F5 8348BE58 6478F631
  7D44152F 494AEBCC A507FA6B 408D6BBB FAAB0A7A 2E7546A8 CA70F9A6 0F7F6824
  554BD833 060D657D ABDF406C 69EEF449 7A4F9AFE 6F0852E7 05DEDAC1 D433191E
  712868C2 A94E642B 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 16801448
  71D9002C 66DFD85F ACB845D1 D25FEA35 74559130 1D060355 1D0E0416 04144871
  D9002C66 DFD85FAC B845D1D2 5FEA3574 5591300D 06092A86 4886F70D 01010405
  00038181 00AFC36B 8A917284 06BD51CB 83BDC4E8 9457A361 6CAAF416 3BBEF691
  04215AC5 EDBC5730 C071C2FB 8A6C90CF D6AB39C2 3BC2147F D35553D9 028B2155
  802E50DB 48CDE067 B3857447 89A1C733 D81EFEF7 1115480F 70ED2F22 F27E35A1
  F3BB597C 7C8F717B FAAD79D3 0F469702 DE9190E4 B1B0808E 46A118EB 887CEAEB
  DFE2900E D2
        quit
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
```

```
!
crypto ikev2 profile prof
 match fvrf any
 match certificate cmap-1
 identity local dn
 authentication local rsa-sig
 authentication remote pre-share
 authentication remote rsa-sig
 pki trustpoint ca-server
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.227 255.255.255.224
 crypto map cmap
!
interface Ethernet1/0
 ip address 209.165.200.228 255.255.255.224
!
ip route 209.165.200.229 255.255.255.224 209.265.200.231
!
ip access-list extended ikev2list
 permit ip any any
!
```

The responder configuration is as follows:

```
crypto pki trustpoint ca-server
 enrollment url http://10.1.1.3:80
 revocation-check none
!
!
!
crypto pki certificate map cmap-2 1
 subject-name eq hostname = initiator
!
crypto pki certificate chain ca-server
 certificate 03
  308201AF 30820118 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32353231 325A170D 31313033 31303132 35323132 5A301A31 18301606 092A8648
  86F70D01 09021609 52455350 4F4E4445 52305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100B517 EB8E64E1 B58CB014 07B3A6AF E6B69577 87486367
  9471B1DA BC66B847 DFA5073A 82121332 E787EA2D 3C433514 39033074 4095E7C7
  67A387A1 EBD24692 A76F0203 010001A3 4F304D30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 144871D9 002C66DF D85FACB8 45D1D25F EA357455
  91301D06 03551D0E 04160414 DFF2401C 53276D96 89DE8C0A 786CCA71 C9EA792B
  300D0609 2A864886 F70D0101 04050003 8181002C 6E334273 CB832A95 3DDC6293
  669E416C A134D543 20952BC3 14A5C0B0 03AE011C 963AF523 C7C5C935 4FE9B2A5
  F24B3161 4D0D723A FA428BD1 85ADF172 B4007067 43C27D8A 1F74ED3D DEBE9F73
  1F515355 E77E766C AEACC303 39457991 29AB090C 99E21B5B 60DCB2C8 780B4479
  3EB3D46B B66C8C26 15311A7A B7A4ED97 32727C
        quit
 certificate ca 01
```

```
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32343933 385A170D 31333033 30393132 34393338 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 DA4ECE09 B998F670 598F32C1 7E9FA920 1D217AC4 293B842E
  7563CE11 B2F0F822 23077930 636C8293 00F6CFDD F6C9B0F5 8348BE58 6478F631
  7D44152F 494AEBCC A507FA6B 408D6BBB FAAB0A7A 2E7546A8 CA70F9A6 0F7F6824
  554BD833 060D657D ABDF406C 69EEF449 7A4F9AFE 6F0852E7 05DEDAC1 D433191E
  712868C2 A94E642B 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 16801448
  71D9002C 66DFD85F ACB845D1 D25FEA35 74559130 1D060355 1D0E0416 04144871
  D9002C66 DFD85FAC B845D1D2 5FEA3574 5591300D 06092A86 4886F70D 01010405
  00038181 00AFC36B 8A917284 06BD51CB 83BDC4E8 9457A361 6CAAF416 3BBEF691
  04215AC5 EDBC5730 C071C2FB 8A6C90CF D6AB39C2 3BC2147F D35553D9 028B2155
  802E50DB 48CDE067 B3857447 89A1C733 D81EFEF7 1115480F 70ED2F22 F27E35A1
  F3BB597C 7C8F717B FAAD79D3 0F469702 DE9190E4 B1B0808E 46A118EB 887CEAEB
  DFE2900E D2
        quit
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
!
crypto ikev2 profile prof
 match fvrf any
 match certificate cmap-2
 identity local dn
 authentication local rsa-sig
 authentication remote pre-share
 authentication remote rsa-sig
 pki trustpoint ca-server
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto dynamic-map dmap 1
 set transform-set trans
 set ikev2-profile prof
!
!
crypto map cmap 1 ipsec-isakmp dynamic dmap
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
 crypto map cmap
!
interface Ethernet1/0
 ip address 209.165.200.232 255.255.255.224
!
ip route 209.165.200.233 255.255.255.224 209.165.200.228
!
ip access-list extended ikev2list
 permit ip host 209.165.200.231 host 209.165.200.228
```

The CA server configuration is as follows:

```
crypto pki server ca-server
```

```
 grant auto
!
crypto pki trustpoint ca-server
 revocation-check crl
 rsakeypair ca-server
!
!
crypto pki certificate chain ca-server
 certificate ca 01
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D303930 33303831
  36333335 395A170D 31323033 30373136 33333539 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 99750598 EF4AF8B4 823DEF66 2F3BBA31 81C2DC5F D9B4040B
  99FB6020 22243CD6 B9F24C84 A543D7DB DD0B3018 2E36208C D0FD4015 EAF0DA69
  C1B0302B 87CEC34B 8646593F 0185AF02 0B86A3F3 5E5C3880 A992CD4A 79F13403
  411CC61F 07CEB4D9 0E967CB2 FAE0A899 5A3B6C87 73111F06 128465DA A45291F8
  F828C5DC 657487E7 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 1680147B
  D032BFB7 B3F70F1A 597B7C1E 1B42E472 5CCD6030 1D060355 1D0E0416 04147BD0
  32BFB7B3 F70F1A59 7B7C1E1B 42E4725C CD60300D 06092A86 4886F70D 01010405
  00038181 003838FA 628804EF E9FF69D9 3D5E299C 29074B2C AE33A563 8AF75976
  78FB68D4 5EF1E27B 04936FDF 78A09432 5348849D F79E17F5 70B233C9 2C1535D0
  506F0C35 99335012 84BBA3DC 050FD3C9 6E7B1D63 41ACC2B5 2B02432D BA2CC2CF
  E379DEA0 A9C208AC 0BEBB2D8 E6488815 EB12F1E0 19072D55 D5D11A49 739144D8
  271A842E ED
        quit
!
interface Ethernet1/0
 ip address 209.165.200.232 255.255.255.224
!
ip http server
```

To obtain the CA and device certificates, enter the **crypto pki authenticate ca-server** and **crypto pki enroll ca-server** commands. To initiate a connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
```

The output of the command is as follows:

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** commands in the responder's CLI to display the session details:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 1.1.1.1 port 500
  IKEv2 SA: local 209.165.200.231/500 remote 209.165.200.227/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 host 209.165.200.226
        Active SAs: 2, origin: dynamic crypto map
```

```
show crypto ikev2 sa detailed
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1         209.165.200.231/500 209.165.200.227/500  (none)/(none)                  READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: RSA, Auth verify: RSA

      Life/Active Time: 86400/846 sec
      CE id: 1001, Session-id: 1
      Status Description: Negotiation done
      Local spi: F79756E978ED41C7       Remote spi: 188FB9A119516D34
      Local id: hostname=RESPONDER
      Remote id: hostname=INITIATOR
      Local req msg id:  0              Remote req msg id:  2
      Local next msg id: 0              Remote next msg id: 2
      Local req queued:  0              Remote req queued:  2
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring Crypto Map- and dVTI-Based IKEv2 Peers

The following example shows how to configure crypto map-and dVTI-based IKEv2 peers using the preshared key authentication method between a static crypto map IKEv2 initiator and a dVTI-based IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 0.0.0.0 0.0.0.0
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 206.165.200.235
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 206.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 206.165.200.227 255.255.255.224
```

```
 crypto map cmap
!
ip route 206.165.200.229 255.255.255.224 206.165.200.235
!
ip access-list extended ikev2list
 permit ip host 206.165.200.227 host 206.165.200.235
 permit ip 206.165.200.233 255.255.255.224 206.165.200.229 255.255.255.224
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer cisco
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
 virtual-template 1
!
crypto ipsec transform-set set esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set ikev2-profile prof
!
interface Loopback0
 ip address 206.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 206.165.200.235 255.255.255.224
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 ip mtu 1000
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
!
```

To initiate a connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 206.165.200.230 source 206.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 206.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 206.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 206.165.200.226-206.165.200.226 Protocol: 1 Port Range: 0-65535;
```

```
 remote traffic selector = Address Range: 206.165.200.230-206.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** command in an Easy VPN server to display the session details:

```
show crypto session
Crypto session current status
Interface: Virtual-Access2
Session status: UP-ACTIVE
Peer: 206.165.200.227 port 500
  IKEv2 SA: local 206.165.200.235/500 remote 206.165.200.227/500 Active
  IPSEC FLOW: permit ip 206.165.200.229/255.255.255.224 206.165.200.233/255.255.255.224
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detail
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   206.165.200.235/500      206.165.200.227/500       (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp: 14, Auth sign: PSK, Auth verify:
PSK
      Life/Active Time: 86400/8 sec
      CE id: 1001, Session-id: 1
      Status Description: Negotiation done
      Local spi: 305F610F57428834       Remote spi: D9D183B5689AEDCD
      Local id: 206.165.200.235
      Remote id: 206.165.200.227
      Local req msg id:  0              Remote req msg id:  2
      Local next msg id: 0              Remote next msg id: 2
      Local req queued:  0              Remote req queued:  2
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
show crypto route
VPN Routing Table: Shows RRI and VTI created routes
Codes: RRI - Reverse-Route, VTI- Virtual Tunnel Interface
        S - Static Map ACLs
Routes created in table GLOBAL DEFAULT
206.165.200.233/255.255.255.224 [1/0] via 206.165.200.227 tag 0
                                on Virtual-Access2 RRI
```

Example: Configuring IPsec Using sVTI-Based IKEv2 Peers

The following example shows how to configure IPsec using the preshared key authentication method between an sVTI IKEv2 initiator and an sVTI IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.225
  pre-shared-key abc
 !
```

```
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 209.165.200.231 255.255.255.224
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set ikev2-profile prof
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Tunnel0
 ip address 10.0.0.1 255.255.255.0
 tunnel source 209.165.200.231
 tunnel mode ipsec ipv4
 tunnel destination 209.165.200.225
 tunnel protection ipsec profile ipsecprof
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
!
ip route 209.165.200.229 255.255.255.224 Tunnel0
!
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.231
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 209.165.200.231 255.255.255.224
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set ikev2-profile prof
!
```

```
crypto map cmap 1 ipsec-isakmp dynamic dmap
!
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Tunnel0
 ip address 10.0.0.2 255.255.255.0
 tunnel source 209.165.200.225
 tunnel mode ipsec ipv4
 tunnel destination 209.165.200.231
 tunnel protection ipsec profile ipsecprof
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
!
ip route 209.165.200.233 255.255.255.224 Tunnel0
```

With sVTI on IKEv2 peers, the session is initiated only when the sVTI interfaces are enabled. In other words, network traffic is not required to initiate the session. To verify the traffic between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.23 Protocol: 1 Port
Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** command in the initiator's CLI to display the session details:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 209.165.200.225 port 500
  IKEv2 SA: local 209.165.200.231/500 remote 209.165.200.225/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detailed
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   209.165.200.231/500  209.165.200.225/500           (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp: 14, Auth sign: PSK, Auth verify:
PSK
      Life/Active Time: 86400/21 sec
      CE id: 1002, Session-id: 2
      Status Description: Negotiation done
      Local spi: 687752902752A6FD       Remote spi: C9DCCFC65493D14F
      Local id: smap-initiator
      Remote id: dmap-responder
      Local req msg id:  2              Remote req msg id:  0
      Local next msg id: 2              Remote next msg id: 0
      Local req queued:  2              Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring IKEv2 on DMVPN Networks

DMVPN uses a tunnel protection CLI that is identical between IKEv1 and IKEv2. The IPsec profile applied on a DMVPN tunnel only refers to an IKEv2 profile. The DMVPN Hub configuration is as follows:

```
crypto ikev2 keyring cisco-ikev2-keyring
 peer dmvpn-node
 description symmetric pre-shared key for the hub/spoke
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco123
crypto ikev2 profile cisco-ikev2-profile
 keyring cisco-ikev2-keyring
 authentication pre-shared
 match local address 0.0.0.0
crypto ipsec profile cisco-ipsec-ikev2
 set transform-set cisco-ts
 set ikev2-profile cisco-ikev2-profile
! interface Tunnel 0
 description This is the Legacy IKEv1 facing tunnel on the hub
 ip address 1.1.1.99 255.255.255.0
 no ip redirects
 ip nhrp map multicast dynamic
 ip nhrp network-id 99
 ip nhrp redirect
 no ip split-horizon eigrp 1
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec
 !
interface Tunnel1
 description This would be the new IKEv2 facing tunnel on the hub
 ip address 2.2.2.99 255.255.255.0
 no ip redirects
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 no ip split-horizon eigrp 1
 tunnel source Ethernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec-ikev2
```

The IKEv2 configuration is as follows:

```
crypto ikev2 profile cisco-ikev2-profile
 keyring cisco-ikev2-keyring
 authentication pre-shared
 match local address 0.0.0.0
crypto ipsec profile cisco-ipsec-ikev2
 set transform-set cisco-ts
 set ikev2-profile cisco-ikev2-profile
interface Tunnel1
 ip address 2.2.2.11 255.255.255.0
 no ip redirects
 ip nhrp map 2.2.2.99 22.22.22.99
 ip nhrp map multicast 22.22.22.99
 ip nhrp network-id 100 ? Keep this same for all IKEv2 spokes for clarity
 ip nhrp nhs 2.2.2.99 ? This points to the hub's IKEv2 facing interface
 tunnel source Ethernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec-ikev2
```

PART XXII

Cisco Group Encrypted Transport VPN

CHAPTER 219

Cisco Group Encrypted Transport VPN

Cisco Group Encrypted Transport VPN (GET VPN) is a set of features that are necessary to secure IP multicast group traffic or unicast traffic over a private WAN that originates on or flows through a Cisco IOS device. GET VPN combines the keying protocol Group Domain of Interpretation (GDOI) with IP security (IPsec) encryption to provide users with an efficient method to secure IP multicast traffic or unicast traffic. GET VPN enables the router to apply encryption to nontunneled (that is, "native") IP multicast and unicast packets and eliminates the requirement to configure tunnels to protect multicast and unicast traffic.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

This document describes how to configure, verify, and troubleshoot Cisco GET VPN.

Cisco Group Encrypted Transport VPN provides the following benefits:

- Provides data security and transport authentication, helping to meet security compliance and internal regulation by encrypting all WAN traffic
- Enables high-scale network meshes and eliminates complex peer-to-peer key management with group encryption keys
- For Multiprotocol Label Switching (MPLS) networks, maintains network intelligence such as full-mesh connectivity, natural routing path, and quality of service (QoS)
- Grants easy membership control with a centralized key server
- Helps ensure low latency and jitter by enabling full-time, direct communications between sites, without requiring transport through a central hub
- Reduces traffic loads on customer premises equipment (CPE) and provider-edge (PE) encryption devices by using the core network for replication of multicast traffic, avoiding packet replication at each individual peer site

Prerequisites for Cisco Group Encrypted Transport VPN

- You must be using Cisco IOS XE Release 2.3 or later.
- You should be knowledgeable about IPsec and Internet Key Exchange (IKE).
- You should know how to configure multicast and unicast routing on a Cisco IOS XE global router.
- When the IKE policy is configured, the IKE lifetime should be set to the minimum of 5 minutes so that unnecessary resources are not wasted on the maintenance of the IKE security association (SA). After the registration IKE SA is established, the registration SAs no longer have to be maintained because the rekey SA has been created and will be used to accept future rekeys.
- When the group rekey lifetime is configured with 300 seconds and forced rekey with policy change is perfomed, you might face network issues. To overcome this issue, one of the following is recommended for group rekey (KEK):
  - Set the lifetime to three times of TEK lifetime configured in transform-set.
  - Set the group rekey lifetime to default value, which is 24 hours (86400 seconds)
  - Configure rekey lifetime as 7200 seconds (2 hours)

Restrictions for Cisco Group Encrypted Transport VPN

- If you are encrypting high packet rates for counter-based antireplay, ensure that you do not make the lifetime too long or it can take several hours for the sequence number to wrap. For example, if the packet rate is 100 kilopackets per second, the lifetime should be configured as fewer than 11.93 hours so that the SA is used before the sequence number wraps.
- Cisco ASR 1000 Series Aggregation Routers with virtual-ppp interface cannot be configured as GETVPN group member.
- In Cisco IOS XE software, an inclusive port range for users to access a network cannot be matched in the extended ACL using the **permit** command.
- For unicast traffic and counter-based antireplay, the sequence numbers may be out of sync between the group members if one of the group members goes down and comes back up. For example: There is traffic from group member 1 to group member 2, and the last sequence number is *n* . Group member 1 goes down and comes back up. The sequence number of the SA at group member 1 now starts with 1, but group member 2 is expecting continuation from the previous sequence number ($n + 1$). This situation causes subsequent traffic from group member 1 to be dropped until the sequence number on group member 1 reaches *n* or the next rekey.
- When you configure transport mode traffic selectors, it is possible to have transport mode SAs. SAs occur when the packet size exceeds the MTU, and the packet cannot be forwarded.
- Transport mode should be used only for Group Encrypted Transport VPN Mode (GM) to GM traffic.

- If you are overriding the don’t fragment bit (df-bit) setting in the IP header of encapsulated packets, you must configure the override commands in global configuration mode. GET VPN does not honor the interface configuration. This restriction is limited only to GET VPN. IPsec accepts both global configuration- and interface-specific override commands.
- Counter-based antireplay is not recommended and works only if there are two group members in a group.
- The GET VPN Time-Based Anti-Replay feature does not support Encapsulating Security Payload (ESP) transport mode in Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4330 Integrated Services Router.
- Because Path MTU Discovery (PMTUD) does not work for GET VPN, there is a possibility that encapsulated packets could be dropped when the df-bit is set and the MTU of an intermediate link is less than the size of the encapsulated packet. In such an event, the router that drops the packet sends a notification to the source IP address on the packet, indicating that the packet has been dropped because the router could not fragment the packet due to the df-bit setting. In GET VPN, this message goes past the encapsulating endpoint directly to the source of the data due to the header preservation feature of GET VPN. Thus, the encapsulating router never knows that it has to fragment the packet to a smaller size before setting the df-bit after encapsulation. It continues to set the df-bit on the packets and they continue to be dropped at the intermediate router. (This is known as null-routing the traffic.)
- In Cisco IOS XE Release 3.5S and earlier releases, key servers cannot be configured using Cisco IOS XE images. They must be configured using Cisco IOS T-based or mainline-based images. This is not a restriction in Cisco IOS XE Release 3.6S and newer releases.
- Because of crypto engine optimization, the time-based antireplay (TBAR) overhead is 16 bytes instead of 12 bytes.
- GET VPN uses TBAR Cisco Metadata Protocol to carry TBAR information. Cisco IOS software uses 12-byte header and Cisco IOS XE uses 16-byte header. Cisco IOS XE software configured on GETVPN group members and using TBAR for anti-replay will have an effective mtu ("cleartext mtu") of the ipsec traffic as 4 bytes lower than group members that configured with Cisco IOS software. When migrating GET VPN group member from Cisco IOS software to Cisco IOS XE software, the reduction in the 4 bytes might result in unexpected performance issues.
- To ensure normal traffic flow for a GET VPN configuration on Cisco ASR 1000 Series Aggregation Services Routers, a TBAR window size greater than 20 seconds is recommended in Cisco IOS XE Release 3.12S and earlier releases, Cisco IOS XE Release 3.14S and Cisco IOS XE Release 3.15S. In Cisco IOS XE Release 3.13S, Cisco IOS XE Release 3.16S and later releases, a TBAR window size lesser than 20 seconds is permitted.
- Crypto maps are not supported on tunnel interface and port-channel interface. However, as an exception to the rule, crypto map for GDOI is supported on tunnel interfaces.
- Crypto maps are not supported on VLAN interfaces.
- RSVP as used in Mediatrace sets the "Router Alert" IP option flag. The Cavium N2 crypto accelerator does not support the use of IP options. Therefore, Mediatrace will fail with IPsec encryption on ASR1000 with Cavium N2. Mediatrace will fail with GETVPN encryption (IPSec with header preservation) on ASR1000 with Cavium N2.
- Deny statements can only be added locally to a GM. Permit statements are not supported in locally configured policies. In case of a conflict, a local policy overrides the policy downloaded from a KS.
- In Cisco ASR 1000 Series Aggregation Services Routers, when there is a failure to reregister, the outbound flow from QFP is not removed since a dummy ACE is pushed instead of a real ACE. As a result, when

the SA expires, the GM will continue to encrypt outbound traffic using an expired SPI, instead of dropping the traffic locally. The traffic eventually gets dropped on the receiving GM due to an invalid SPI mechanism.

- While configuring an IPv6 access list on a Key Server, do not use the **ahp** option with the **permit** or **deny** commands.
- A Cisco IOS XE platform running as a GETVPN group member can only support one GETVPN-ipv4 group member instance and one GETVPN-ipv6 group member instance.
- **SSO Restrictions**
- Cisco ASR 1000 Series Routers support stateful IPsec sessions on Embedded Services Processor (ESP) switchover. During ESP switchover, all IPsec sessions will stay up and no user intervention is needed to maintain IPsec sessions.
- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.
- The Cisco ASR 1000 Series Router currently does not support Stateful Switchover (SSO) IPsec sessions on Route Processors (RPs). The IPsec sessions will go down on initiation of the switchover, but will come back up when the new RP becomes active. No user intervention is needed. Traffic disruption might happen over the IPSec sessions for the duration of the switchover, until the sessions are back up.
- Cisco ASR 1000 Series Router does not support stateful ISSU for IPsec sessions. Before performing an ISSU, you must explicitly terminate all existing IPSec sessions or tunnels prior to the operation and reestablish them post ISSU. Specifically, ensure that there are no half-open or half-established IPsec tunnels present before performing ISSU. To do this, we recommend a interface shutdown in the case of interfaces that may initiate a tunnel setup, such as a routing protocol initiating a tunnel setup, or interfaces that have keepalive enabled, or where there is an auto trigger for an IPsec session. Traffic disruption over the IPsec sessions during ISSU is obvious in this case.

Information About Cisco Group Encrypted Transport VPN

Cisco Group Encrypted Transport VPN Overview

Networked applications such as voice and video increase the need for instantaneous, branch-interconnected, and QoS-enabled WANs. The distributed nature of these applications results in increased demands for scale. At the same time, enterprise WAN technologies force businesses to trade off between QoS-enabled branch interconnectivity and transport security. As network security risks increase and regulatory compliance becomes essential, GET VPN, a next-generation WAN encryption technology, eliminates the need to compromise between network intelligence and data privacy.

With GET, Cisco provides tunnelless VPN, which eliminates the need for tunnels. Meshed networks, by removing the need for point-to-point tunnels, can scale higher while maintaining network-intelligence features critical to voice and video quality. GET is a standards-based security model that is based on the concept of “trusted” group members. Trusted member routers use a common security methodology that is independent of any point-to-point IPsec tunnel relationship. Also, “any-any” networks, by using trusted groups instead of

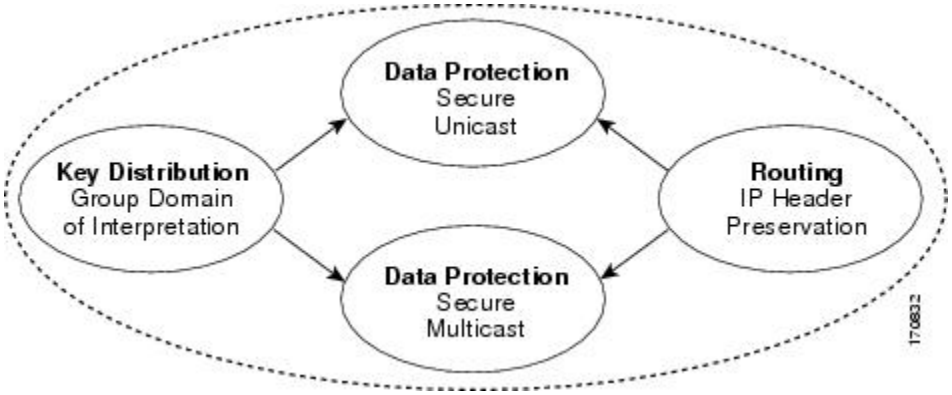
point-to-point tunnels, can scale higher while maintaining network-intelligence features (such as QoS, routing, and multicast), which are critical to voice and video quality.

GET-based networks can be used in a variety of WAN environments, including IP and MPLS. MPLS VPNs that use this encryption technology are highly scalable, manageable, and cost-effective, and they meet government-mandated encryption requirements. The flexible nature of GET allows security-conscious enterprises either to manage their own network security over a service provider WAN service or to offload encryption services to their providers. GET simplifies securing large Layer 2 or MPLS networks that require partial or full-mesh connectivity.

Cisco Group Encrypted Transport VPN Architecture

GET VPN encompasses Multicast Rekeying, a way to enable encryption for "native" multicast packets, and unicast rekeying over a private WAN. Multicast Rekeying and GET VPN is based on GDOI as defined in Internet Engineering Task Force (IETF) RFC 3547. In addition, there are similarities to IPsec in the area of header preservation and SA lookup. Dynamic distribution of IPsec SAs has been added, and tunnel overlay properties of IPsec have been removed. The figure below further illustrates the GET VPN concepts and relationships.

Figure 113: GET VPN Concepts and Relationships



Key Distribution Group Domain of Interpretation

GDOI

GDOI is defined as the Internet Security Association Key Management Protocol (ISAKMP) Domain of Interpretation (DOI) for group key management. In a group management model, the GDOI protocol operates between a group member and a group controller or key server (GCKS), which establishes SAs among authorized group members. The ISAKMP defines two phases of negotiation. GDOI is protected by a Phase 1 ISAKMP security association. The Phase 2 exchange is defined in RFC 6407. The topology shown in the figure below and the corresponding explanation show how this protocol works.

Group Member

The group member registers with the key server to get the IPsec SA or SAs that are necessary to communicate with the group. The group member provides the group ID to the key server to get the respective policy and keys for this group. These keys are refreshed periodically, and before the current IPsec SAs expire, so that there is no loss of traffic.

The output of the **show crypto isakmp sa detail** command will show the security association (SA) Authentication as "rsig" because the RSA signature is used for key encryption key (KEK) rekey authentication in GET VPN.

Key Server

The responsibilities of the key server include maintaining the policy and creating and maintaining the keys for the group. When a group member registers, the key server downloads this policy and the keys to the group member. The key server also rekeys the group before existing keys expire.

Note In Cisco IOS XE Release 3.5S and earlier releases, key servers are not supported on the Cisco ASR 1000 series routers. They must be configured using Cisco IOS T-based or mainline-based images. This is not a restriction on Cisco IOS XE Release 3.6S and newer releases.

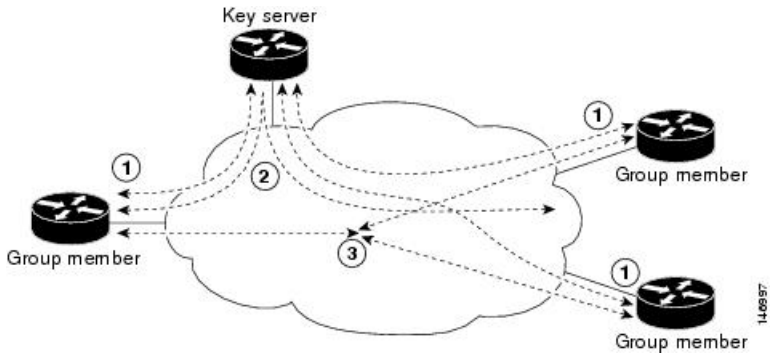
The key server has two responsibilities: servicing registration requests and sending rekeys. A group member can register at any time and receive the most current policy and keys. When a group member registers with the key server, the key server verifies the group ID that the group member is attempting to join. If this ID is a valid group ID, the key server sends the SA policy to the group member. After the group member acknowledges that it can handle the downloaded policy, the key server downloads the respective keys.

There are two types of keys that the key server can download: the key encryption key (KEK) and the traffic encryption key (TEK). The TEK becomes the IPsec SA with which the group members within the same group communicate. The KEK encrypts the rekey message.

The GDOI server sends out rekey messages if an impending IPsec SA expiration occurs or if the policy has changed on the key server (using the command-line interface [CLI]). With CSCti89255, KEK rekeys before the KEK timer expires. The group member also starts a timer and expects to receive refreshed keys before timer expiration. If they are not received, the group member initiates a jittered re-registration prior to KEK expiry. KEK is deleted when the KEK lifetime expires.

The rekey messages may also be retransmitted periodically to account for possible packet loss. Packet loss can occur because rekey messages are sent without the use of any reliable transport. If the rekey mechanism is multicast, there is no efficient feedback mechanism by which receivers can indicate that they did not receive a rekey message, so retransmission seeks to bring all receivers up to date. If the rekey mechanism is unicast, the receivers will send an acknowledgment message.

Figure 114: Protocol Flows That Are Necessary for Group Members to Participate in a Group



The topology shows the protocol flows that are necessary for group members to participate in a group, which are as follows:

1. Group members register with the key server. The key server authenticates and authorizes the group members and downloads the IPsec policy and keys that are necessary for them to encrypt and decrypt IP multicast packets.
2. As needed, the key server "pushes" a rekey message to the group members. The rekey message contains a new IPsec policy and keys to use when old IPsec SAs expire. Rekey messages are sent in advance of the SA expiration time to ensure that valid group keys are always available.
3. The group members are authenticated by the key server and communicate with other authenticated group members that are in the same group using the IPsec SAs that the group members have received from the key server.

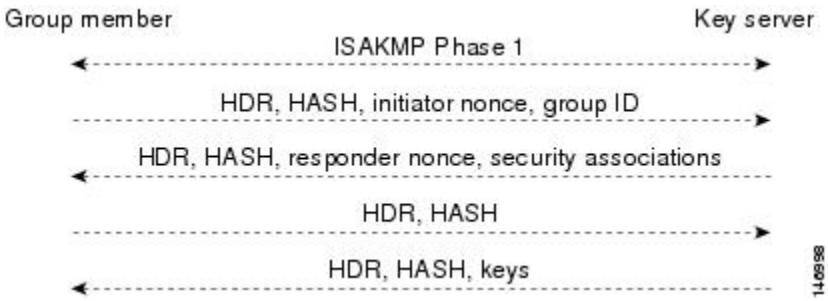
How Protocol Messages Work with Cisco Software

Multicast Rekeying uses the GDOI protocol (RFC 6407) to distribute the policy and keys for the group. The GDOI protocol is between a key server and a group member. The key server creates and maintains the policy and keys, and it downloads the policy and keys to the authenticated group members.

The GDOI protocol is protected by an ISAKMP Phase 1 exchange. The GDOI key server and the GDOI group member must have the same ISAKMP policy. This Phase 1 ISAKMP policy should be strong enough to protect the GDOI protocol that follows. The GDOI protocol is a four-message exchange that follows the Phase 1 ISAKMP policy. The Phase 1 ISAKMP exchange can occur in main mode or aggressive mode.

The figure below shows the ISAKMP Phase 1 exchange.

Figure 115: ISAKMP Phase 1 Exchange and GDOI Registration



The ISAKMP Phase 1 messages and the four GDOI protocol messages are referred to as the GDOI registration, and the entire exchange that is shown is a unicast exchange between the group member and the key server.

During the registration, if the rekey mechanism is multicast, the group member receives the address of the multicast group and registers with the multicast group that is required to receive the multicast rekeys.

The GDOI protocol uses User Datagram Protocol (UDP) port 848 (with Network Address Translation-Traversal (NAT-T), it floats to 4500).

IPsec

IPsec is a well-known RFC that defines an architecture to provide various security services for traffic at the IP layer. The components and how they fit together with each other and into the IP environment are described in IETF RFC 2401.

Communication Flow Between Key Servers and Group Members to Update IPsec SAs

Key servers and group members are the two components of the GET VPN architecture. The key server holds and supplies group authentication keys and IPsec SAs to the group members.

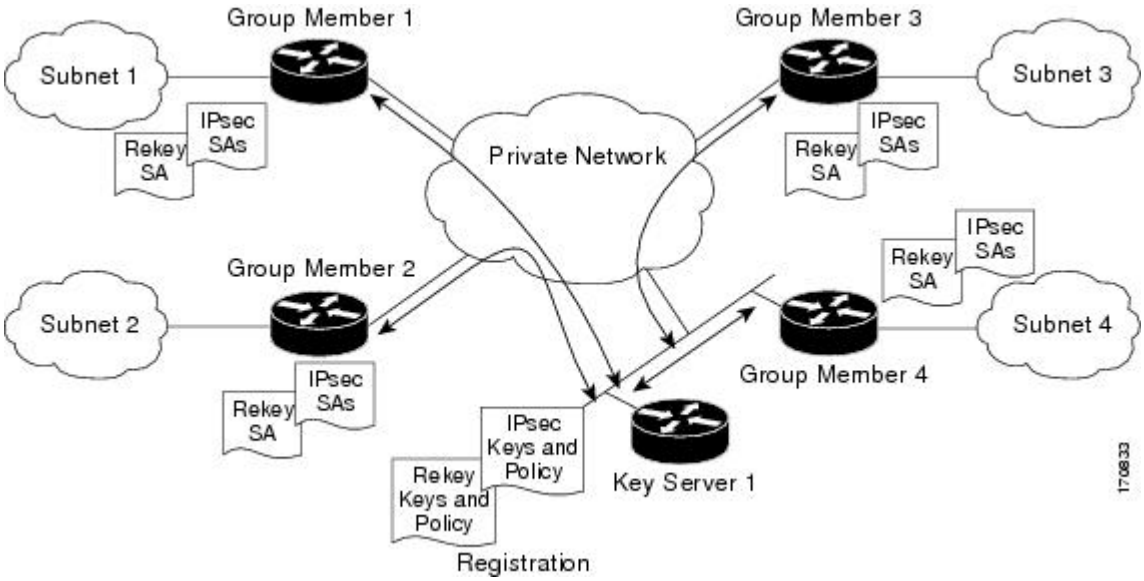
Group members provide encryption service to the interesting traffic (traffic that is worthy of being encrypted and secured by IPsec).

Communication among the key server and group members is encrypted and secured. GDOI supports the use of two keys: the TEK and the KEK. The TEK is downloaded by the key server to all the group members. The downloaded TEK is used by all the group members to communicate securely among each other. This key is essentially the group key that is shared by all the group members. The group policies and IPsec SAs are refreshed by the key server using periodic rekey messages to the group members. The KEK is also downloaded by the key server and is used by the group members to decrypt the incoming rekey messages from the key server.

The key server generates the group policy and IPsec SAs for the GDOI group. The information generated by the key server includes multiple TEK attributes, traffic encryption policy, lifetime, source and destination, a Security Parameter Index (SPI) ID that is associated with each TEK, and the rekey policy (one KEK).

The figure below illustrates the communication flow between group members and the key server. The key server, after receiving registration messages from a group member, generates the information that contains the group policy and new IPsec SAs. The new IPsec SA is then downloaded to the group member. The key server maintains a table that contains the IP address of each group member per group. When a group member registers, the key server adds its IP address in its associated group table, thus allowing the key server to monitor an active group member. A key server can support multiple groups. A group member can be part of multiple groups.

Figure 116: Communication Flow Between Group Members and the Key Server



IPsec and ISAKMP Timers

IPsec and ISAKMP SAs are maintained by the following timers:

- TEK lifetime-Determines the lifetime of the IPsec SA. Before the end of the TEK lifetime, the key server sends a rekey message, which includes a new TEK encryption key and transforms as well as the existing KEK encryption keys and transforms. The TEK lifetime is configured only on the key server, and the lifetime is "pushed down" to the group members using the GDOI protocol. The TEK lifetime value depends on the security policy of the network. If the **set security-association lifetime** command is not

configured, the default value of 86,400 seconds takes effect. To configure a TEK lifetime, see the "Configuring an IPsec Lifetime Timer" section.

- KEK lifetime-Determines the lifetime of the GET VPN rekey SAs. Before the end of the lifetime, the key server sends a rekey message, which includes a new KEK encryption key and transforms and new TEK encryption keys and transforms. The KEK lifetime is configured only on the key server, and the lifetime is pushed down to group members dynamically using the GDOI protocol. The KEK lifetime value should be greater than the TEK lifetime value (it is recommended that the KEK lifetime value be at least three times greater than the TEK lifetime value). If the **rekey lifetime** command is not configured, the default value of 86,400 seconds takes effect. To configure a KEK lifetime, see the "Configuring a Multicast Rekey" section.
- ISAKMP SA lifetime-Defines how long each ISAKMP SA should exist before it expires. The ISAKMP SA lifetime is configured on a group member and on the key server. If the group members and key servers do not have a cooperative key server, the ISAKMP SA is not used after the group member registration. In this case (no cooperative key server), the ISAKMP SA can have a short lifetime (a minimum of 60 seconds). If there is a cooperative key server, all key servers must have long lifetimes to keep the ISAKMP SA "up" for cooperative key server communications. If the **lifetime** command is not configured, the default value of 86,400 seconds takes effect. To configure an ISAKMP SA lifetime, see the "Configuring an ISAKMP Lifetime Timer" section.

Address Preservation

The following section describes address preservation in GET VPN.

As shown in the figure below, IPsec-protected data packets carry the original source and destination in the outer IP header rather than replacing them with tunnel endpoint addresses. This technique is known as IPsec Tunnel Mode with Address Preservation.

Figure 117: Header Preservation

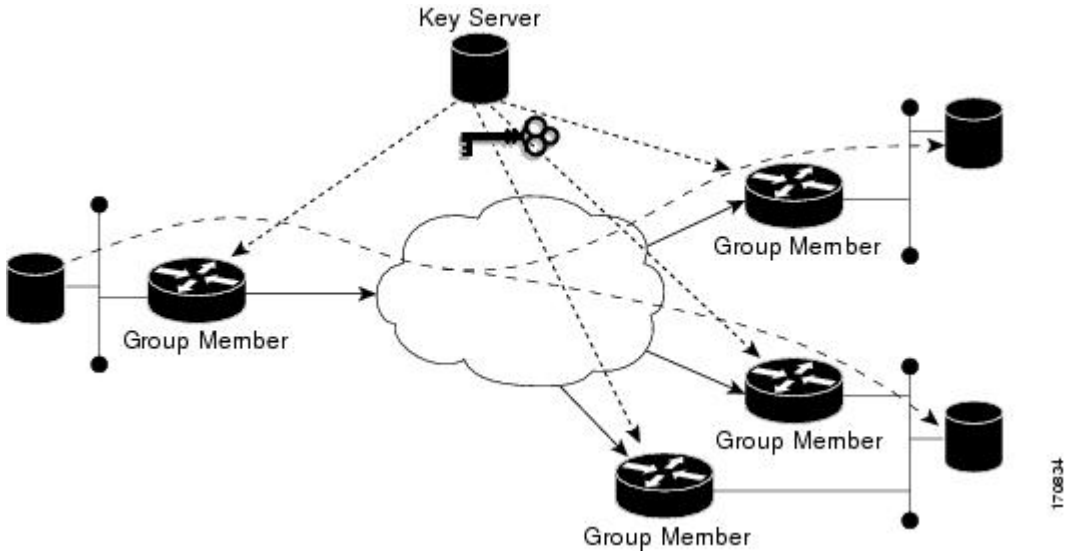
Address preservation allows GET VPN to use the routing functionality present within the core network. Address preservation allows routing to deliver the packets to any customer-edge (CE) device in the network that advertises a route to the destination address. Any source and destination matching the policy for the group will be treated in a similar manner. In the situation where a link between IPsec peers is not available, address preservation also helps combat traffic "route absence" situations.

Header preservation also maintains routing continuity throughout the enterprise address space and in the WAN. As a result, end host addresses of the campus are exposed in the WAN (for MPLS, this applies to the edge of the WAN). For this reason, GET VPN is applicable only when the WAN network acts as a "private" network (for example, in an MPLS network).

Secure Data Plane Multicast

The multicast sender uses the TEK that is obtained from the key server and encrypts the multicast data packet with header preservation before it switches out the packet. The replication of the multicast packet is carried out in the core on the basis of the (S, G) state that is retained in the multicast data packet. This process is illustrated in the figure below.

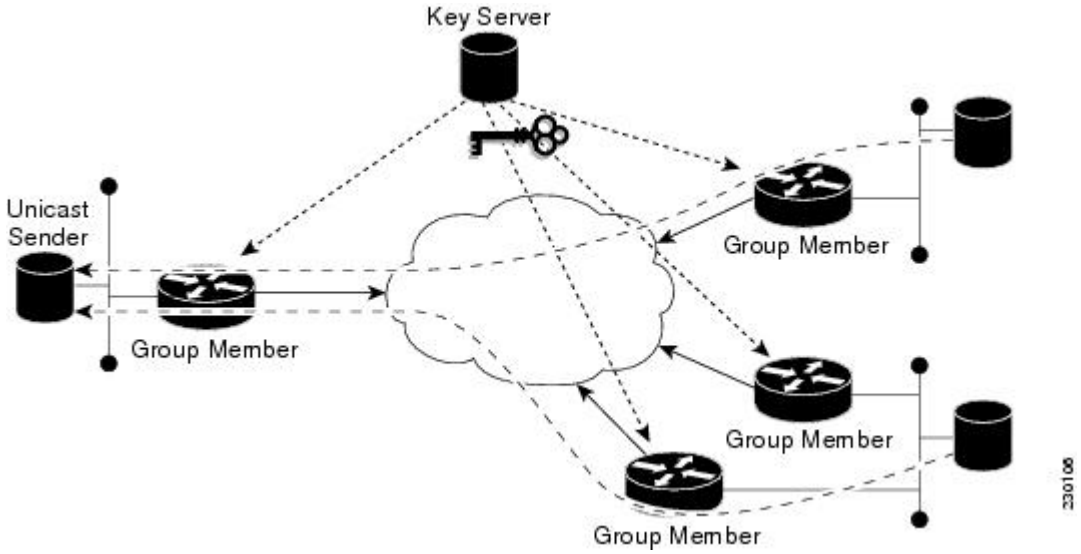
Figure 118: Secure Data Plane Multicast Process



Secure Data Plane Unicast

The unicast sender uses the TEK that is obtained from the key server and encrypts the unicast data packet with header preservation before it switches out the packet to the destination. This process is illustrated in the figure below.

Figure 119: Secure Data Plane Unicast Process



Cisco Group Encrypted Transport VPN Features

Rekeying

Rekey messages are used to refresh IPsec SAs. When the IPsec SAs or the rekey SAs are about to expire, one single rekey message for a particular group is generated on the key server. No new IKE sessions are created for the rekey message distribution. The rekey messages are distributed by the key server over an existing IKE SA.

Rekeying can use multicast or unicast messages. GET VPN supports both unicast and multicast rekeying.

With CSCti89255, KEK rekeys before the KEK timer expires. The group member also starts a timer and expects to receive refreshed keys before timer expiration. If they are not received, the group member initiates a jittered re-registration prior to KEK expiry. KEK is deleted when the KEK lifetime expires. This ensure the following:

- A safer KEK expiry checking mechanism
- A safer KEK re-registration mechanism
- Avoids use of KEK beyond configured lifetime

The following subsections give detailed rekeying information:

Rekey Sequence Number

Before the end of a TEK/KEK lifetime, KS sends a rekey message with the sequence number incremented by 1. However, if a secondary KS has become the primary KS in the time since the last rekey message was sent, the new primary KS increments the sequence number of the rekey message by 10.

The primary KS and secondary KS synchronize sequence numbers every 20 seconds.

The following example shows how the sequence number of rekey messages changes in a deployment consisting of a primary KS, KS1, and a secondary KS, KS2. For the sake of the example, we assume the sequence number has the initial value 1.

We also assume that the deployment has a large number of GMs and that the KS may need to retry the delivery of rekey messages. The sequence number is incremented by 1 for each retry.

1. When it is time to send a rekey message, KS1 increments the sequence number to 2.
2. Suppose that KS1 resends the rekey message thrice so that all the GMs receive the message. With each retry, the sequence number is incremented by 1. So, the value of the sequence number at the end of this rekey is 5.
3. When it is time to send the next rekey message, suppose that KS1 sends the rekey message only once. So, the sequence number at the end of this second rekey is 6.
4. Before the next rekey message is sent, suppose KS2 becomes the primary KS.
5. When it is time to send the rekey message, KS2 increments the sequence number by 10. So, the rekey message is sent with the sequence number 16.
6. Suppose that KS2 resends the rekey message twice so that all the GMs receive the message. With each retry, the sequence number is incremented by 1. So, the value of the sequence number at the end of this rekey is 18.
7. Before the next rekey message is sent, suppose that KS1 becomes the primary KS.

8. When it is time to send the rekey message, KS1 increments the sequence number by 10. So, the rekey message is sent with the sequence number 28. Suppose that KS1 sends the rekey message only once. The sequence number at the end of the rekey is 28.
9. When it is time to send the next rekey message, KS1 increments the sequence number by 1. Suppose KS1 sends the rekey message only once. The sequence number at the end of the rekey is 29.

The following table summarizes the change in sequence numbers during each rekey operation:

Rekey #	1(3 retries)	2(0 retries)	3(2 retries	4(0 retries)	5(0 retries)
Sequence #	2,3,4,5	6	16,17,18	28	29

Rekey Sequence-Number Check

The rekey sequence-number check between the key server and the group member is conducted as follows:

1. Antireplay in GROUPKEY-PUSH messages is restored as specified in RFC 6407.
   - The group member drops any rekey message that has a sequence number lower than or equal to that of the last received rekey message.
   - The group member accepts any rekey message that has a sequence number higher than that of the last received rekey message, no matter how large the difference.
2. The sequence number is reset to 1 at the first rekey message after the KEK rekey, not at the KEK rekey message itself.

Multicast Rekeying

Multicast rekeys are sent out using an efficient multicast rekey. Following a successful registration, the group member registers with a particular multicast group. All the group members that are registered to the group receives this multicast rekey. Multicast rekeys are sent out periodically on the basis of the configured lifetime on the key server. Multicast rekeys are also sent out if the IPsec or rekey policy is changed on the key server. Triggered by the configuration change, the rekey sends out the new updated policy to all the group members with an efficient multicast rekey.

The key server pushes the rekey time back as follows:

1. If the TEK timeout is 300 seconds:

tek_rekey_offset = 90 (because 300 < 900)

If retransmissions are configured, the rekey timer is moved back more.

For three retransmissions every 10 seconds: 3 x 10

So the rekey will actually happen at (300 - 90 - 30) = 180 seconds

2. If the TEK timeout is 3600 seconds:

tek_rekey_offset = 3600 x 10 percent = 360 seconds

If retransmissions are configured, the rekey timer is moved back more.

For three retransmissions every 10 seconds: 3 x 10

So the rekey will actually happen at (3600 - 360 - 30) = 3210 seconds

When a KEK expires and when the transport mode is multicast, a multicast KEK rekey is sent. When a multicast KEK rekey is sent, the group member replaces the old KEK with the new KEK. Because it is a

multicast rekey, and the retransmissions are sent, the old KEK continues to be used for encryption. This situation occurs because the group member does not receive the new KEK rekey. Hence the group member that received the multicast KEK rekey does not have the old KEK, and hence it drops these retransmissions.

The group member that did not initially receive the KEK key now receives the KEK retransmission and replaces the old KEK with the new KEK and will drop the retransmissions that will follow. For example, if five retransmissions are configured and a multicast KEK rekey with sequence number 1 is received at group member 1, all the other retransmissions with sequence numbers 2 3 4 5 6 will be dropped at the group member because the group member does not have the old KEK.

If group member 2 does not get the KEK rekey with sequence number 1 and it receives the retransmission with sequence number 2, it will drop the other retransmissions 3, 4, 5, 6.

Configuration Requirements for Multicast Rekeying

When a group member registers to a key server, it installs the KEK SA into its database. When the rekey transport is multicast the group member will use IGMP to join the multicast stream defined by the key server. The IGMP join is transmitted from the interface that contains the crypto map.

Note The IGMP traffic should be excluded from encryption via either the ACL defined on the key server or a local deny ACL on the group member.

When the key server is not reachable via the same interface as the one configured with the crypto map, it will have to manually join the stream.

Unicast Rekeying and SAs

In a large unicast group, to alleviate latency issues, the key server generates rekey messages for only a small number of group members at a time. The key server is ensured that all group members receive the same rekey messages for the new SA before the expiration of the old SA. Also, in a unicast group, after receiving the rekey message from the key server, a group member sends an encrypted acknowledge (ACK) message to the key server using the keys that were received as part of the rekey message. When the key server receives this ACK message, it notes this receipt in its associated group table, which accomplishes the following:

- The key server keeps a current list of active group members.
- The key server sends rekey messages only to active members.

In addition, in a unicast group, the key server removes the group member from its active list and stops sending the rekey messages to that particular group member if the key server does not receive an ACK message for three consecutive rekeys. If no ACK message is received for three consecutive rekeys, the group member has to fully re-register with the key server after its current SA expires if the group member is still interested in receiving the rekey messages. The ejection of a nonresponsive group member is accomplished only when the key server is operating in the unicast rekey mode. The key server does not eject group members in the multicast rekey mode because group members cannot send ACK messages in that mode.

As in multicast rekeying, if retransmission is configured, each rekey will be retransmitted the configured number of times.

Rekey transport modes and authentication can be configured under a GDOI group.

If unicast rekey transport mode is not defined, multicast is applied by default.

If the TEK rekey is not received, the group member re-registers with the key server 60 seconds before the current IPsec SA expires. The key server has to send out the rekey before the group member re-registration occurs. If no retransmission is configured, the key server sends the rekey tek_rekey_offset before the SA expires. The tek_rekey_offset is calculated based on the configured rekey lifetime. If the TEK rekey lifetime is less than 900 seconds, the tek_rekey_offset is set to 90 seconds. If the TEK rekey lifetime is configured as more than 900 seconds, the tek_rekey_offset = (configured TEK rekey lifetime)/10. If retransmission is configured, the rekey occurs earlier than the tek_rekey_offset to let the last retransmission be sent 90 seconds before the SA expires.

The key server uses the formula in the following example to calculate when to start sending the rekey to all unicast group members. The unicast rekey process on the key server sends rekeys to unicast group members in groups of 50 within a loop. The time spent within this loop is estimated to be 5 seconds.

A key server rekeys group members in groups of 50, which equals two loops. For example, for 100 group members:

Number of rekey loops = (100 group members)/50 = 2 loops:

- Time required to rekey one loop (estimation) = 5 seconds
- Time to rekey 100 group members in two loops of 50: 2 x 5 seconds = 10 seconds

So the key server pushes the rekey time back as follows:

- If the TEK timeout is 300: 300 - 10 = 290

But the start has to be earlier than the TEK expiry (as in the multicast case):

- Because 300 < 900, tek_rekey_offset = 90
- So 90 seconds is subtracted from the actual TEK time: 290 - tek_rekey_offset = 200 seconds

If retransmissions are configured, the rekey timer is moved back more:

- For three retransmissions every 10 seconds: 200 - (3 x 10) = 170
- If the TEK timeout is 3600 seconds: 3600 - 10 = 3590

But the start has to be earlier than the TEK expiry (as in the multicast case):

- Because 3600 > 900, tek_rekey_offset = 3600 x 10 percent = 360
- So 360 seconds is subtracted from the actual TEK time: 3590 - tek_rekey_offset = 3230 seconds

If retransmissions are configured, the rekey timer is moved back more:

- For three retransmissions every 10 seconds: 3230 - (3 x 10) = 3200 seconds

The tek_rekey_offset formula applies to unicast and multicast rekeying.

Rekey Behavior After Policy Changes

The table below provides a list of rekey behavior based on the security policy changes.

Table 286: Rekey Behavior After Security Policy Changes

Policy Changes	Rekey Sent?	Rekey Behavior After Policy Changes
TEK: SA lifetime	No	The old SA remains active until its lifetime expires. The new lifetime will be effective after the next scheduled rekey.
TEK: IPSEC transformset	Yes	The SAs of the old transform set remain active until its lifetime expires.
TEK: IPSEC profile	Yes	The SAs of the old profile remain active until its lifetime expires.
TEK:matching ACL	Yes	Outbound packet classification will use the new access control list (ACL) immediately. The old SAs are still kept in the SA database.
TEK:enable replay counter	Yes	The old SA without counter replay remains active until its lifetime expires.
TEK:change replay counter	No	The SA with a new replay counter will be sent out in the next scheduled rekey.
TEK:disable replay counter	Yes	The old SA with counter replay enabled remains active until its lifetime expires.
TEK:enable receive-only	Yes	Receive-only mode is activated immediately after rekey.
TEK:disable receive-only	Yes	Receive-only mode is deactivated immediately after rekey.
KEK:SA lifetimebehavior	No	Change is applied with the next rekey.
KEK:change authentication key	Yes	Change is applied with the next rekey.
KEK:changing crypto algorithm	Yes	Change is applied immediately.

Enter the following commands for the policy changes to take effect immediately:

- Use the **clear crypto gdoi** [**group**] command on the key server.
- Use the **clear crypto gdoi** [**group**] command on all the group members.

Note The key server sends rekeys for policy updates after the administrator exits configuration mode, ensuring that the rekeys are sent when appropriate.

Note Passive-mode behavior before changing to bidirectional mode on a group member is as follows:

If an you change the SA mode on the key server to "no sa receive-only," and exits configuration mode, the rekey is sent to the group member, and you can see the state on the group member changing from "inbound only" to "inbound optional;" the state will change to "both" after an interval set by a built-in timer; about five minutes.

The key server shows this state as "both" immediately; this is done by design because all group members might be in the process of being updated.

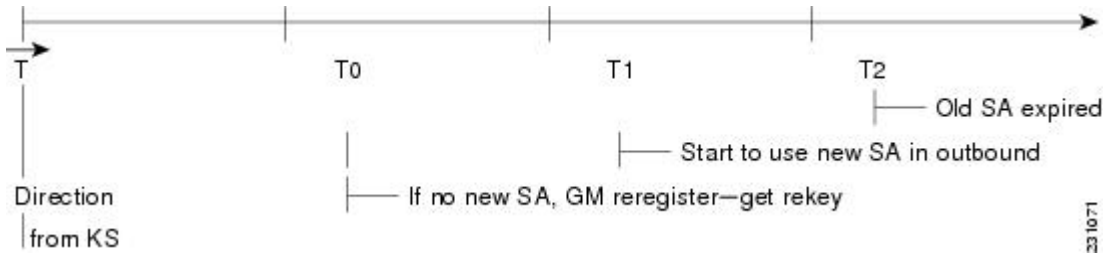
IPsec SA Usage on the Group Members

When a rekey is received and processed on a group member, the new IPsec SA (the SPI) is installed. There is a period of time when the old and the new IPsec SAs are used. After a certain specified interval, the old IPsec SA is deleted. This overlap ensures that all group members receive the current rekey and insert the new IPsec SAs. This behavior is independent of the transport method (multicast or unicast rekey transport) for the rekeys from the key server.

Approximately 30 seconds before the old SA expires, the group member starts to use the new SA in the outbound direction to encrypt the packet. Approximately 60 seconds before the old SA expires, if no new SA is received on the group member side via a rekey from the key server, the group member reregisters.

In the figure below, time T2 is when the old SA expires. T1 is 30 seconds before T2, which is when the group member (GM) starts to use the new SA in the outbound direction. T0 is another 30 seconds before T2. If no new SA is received at T0, the group member has to reregister. T is another 30 seconds from T0. The key server should send a rekey at T.

Figure 120: IPsec SA Usage on a Group Member



Configuration Changes Can Trigger a Rekey By a Key Server

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Configuration changes on a key server can trigger a rekey by the key server. Please refer to the following sample configuration as you read through the changes that will or will not cause a rekey that are described following the example.

```
crypto ipsec transform-set gdoi-p esp-aes esp-sha-hmac
!
```

```
crypto ipsec profile gdoi-p
 set security-association lifetime seconds 900
set transform-set gdoi-p
!
crypto gdoi group diffint
 identity number 3333
 server local
  rekey algorithm aes 128
  rekey address ipv4 121
  rekey lifetime seconds 3600
  no rekey retransmit
  rekey authentication mypubkey rsa mykeys
  sa ipsec 1
   profile gdoi-p
   match address ipv4 120
   replay counter window-size 3
```

The following configuration changes on the key server will trigger a rekey from the key server:

- Any change in the TEK configuration ("sa ipsec 1" in the example):
  - If the ACL ("match address ipv4 120" in the above example) is changed. Any addition, deletion, or change in the ACL causes a rekey.
  - If TEK replay is enabled or disabled on the key server, rekey is sent.
  - Removal or addition of the IPsec profile in the TEK ("profile gdoi-p" in the example).
  - Changing from multicast to unicast transport.
  - Changing from unicast to multicast transport.

The following configuration changes on the key server will not trigger a rekey from the key server:

- Replay counter window size is changed under the TEK ("sa ipsec 1" in the example).
- Configuring or removing rekey retransmit.
- Removing or configuring the rekey ACL.
- Changing the TEK lifetime ("set security-association lifetime seconds 300" in the example) or changing the KEK lifetime ("rekey lifetime seconds 500" in the example).
- Adding, deleting, or changing the rekey algorithm ("rekey algorithm aes 128" in the example).

Commands That Trigger a Rekey

The table below is a comprehensive list of GET VPN command changes, and it shows which commands will or will not trigger a rekey. Commands are broken out based on the configuration mode in which they are entered. The table also shows when the commands take effect, regardless of whether they trigger a rekey.

Note When the KEK lifetime is changed in the GDOI group, the changes take place only when the current KEK expires and a new one is generated. You can force the changes to take place, by issuing the rekey command, **crypto gdoi ks rekey**, on the key server.

Table 287: Commands That Trigger a Rekey

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Mode = (config)	**configure terminal**	—	—	—
Change/delete ACL used in GDOI group (example: **rekey address ipv4** *access-list-number*[*options*])	[**no**] **access-list** *access-list-number*[*options*]	No	—	Immediately
Change/delete ACL used in IPsec profile (example: **match address ipv4** *access-list-id* | *name*[*options*])	[**no**] **access-list** *access-list-number*[*options*]	Yes	End configuration mode	**show running-config** command output on key server indicates that the policy is incomplete, the packet is still encrypted/decrypted by the existing SA, downloaded ACLs are cleared but multidimensional-tree entries are still present (by displaying **show crypto ruleset** command output), and no new SAs are downloaded and old SAs are still active in encrypt/decrypt.
Add/remove ISAKMP preshared key (arbitrary key)	**crypto isakmp key address** *peer-address*	No	—	Immediately
Add/remove ISAKMP preshared key (group member key)	**crypto isakmp key address** *peer-address*	No	—	After key encryption key (KEK) SA expires (re-registration)
Add IPsec profile	**crypto ipsec profile**	No	—	Immediately
Add/remove ISAKMP policy	**crypto isakmp policy** *priority*	No	—	Immediately
Mode = (ipsec-profile)	**crypto ipsec profile** *name*	—	—	—
Change SA lifetime (in IPsec profile)	**set security-association lifetime** *seconds*	No	—	Next rekey
Change transform-set	**set transform-set** *transform-set-name*	Yes	End configuration mode	The SAs of the old transform set remain active until the lifetime expires.

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Mode = (config-gdoi-group)	**crypto gdoi group** *group-name*	—	—	—
Change identity number	**identity number** *number*	No	—	Must immediately configure on the group member. The other group members keep using the TEKs and KEKs of the old group ID.
Mode = (gdoi-local-server)	**server local**	—	—	—
Change from unicast to multicast transport	**rekey transport unicast**	Yes	Immediately	After triggered rekey
Change from multicast to unicast transport	[**no**] **rekey transport unicast**	Yes	End configuration mode	After triggered rekey
Change rekey address	**rekey address ipv4** {*access-list-number* | *access-list-name*}	Yes	End configuration mode	After triggered rekey (however, changing the ACL itself will not trigger a multicast rekey)
Change rekey lifetime	**rekey lifetime seconds** *number-of-seconds*	No	—	Next rekey, but lifetime starts decrementing when the command is issued (the current lifetime is sent out with the rekey).
Enable/disable rekey retransmit	**rekey retransmit** *number-of-seconds* [**number** *number-of-retransmissions*]	No	—	Next rekey
Enable rekey authentication	**rekey authentication mypubkey rsa** *key-name*	Yes	End configuration mode	After triggered rekey
Disable rekey authentication	[**no**] **rekey authentication**	No	—	Immediately
Change rekey authentication key	**rekey authentication mypubkey rsa** *key-name*	Yes	End configuration mode	After triggered rekey
Change rekey encryption	**rekey algorithm** *type-of-encryption-algorithm*	Yes	End configuration mode	New algorithm takes effect immediately.
Mode = (gdoi-sa-ipsec)	**sa ipsec** *sequence-number*	—	—	—

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Change profile	**profile** *ipsec-profile-name*	Yes	End configuration mode	SAs of the old profile are still in effect until the lifetime expires.
Change ACL match	**match address** [options]	Yes	End configuration mode	After triggered rekey
Enable counter replay	**replay counter window-size** *seconds*	Yes	End configuration mode	Old SA without counter replay is still inactive until the lifetime expires.
Change replay counter value	**replay counter window-size** *seconds*	No	—	Next rekey
Enable time-based antireplay	**replay time window-size** *seconds*	Yes	End configuration mode	New SA with time-based antireplay enabled is sent, but the old SA with time-based antireplay disabled is still active until the lifetime expires.
Change time-based antireplay window	**replay time window-size** *seconds*	No	—	New time-based antireplay window is effective only after entering the **clear crypto gdoi** command on both the key server and group member.
Mode = (gdoi-coop-ks-config)	**redundancy**	—	—	—
Enable redundancy	**redundancy**	No	—	Must immediately configure on other key servers
Change local priority	**local priority** *number*	No	—	Immediately but does not force key server election
Add/remove peer address	[**no**] **peer address ipv4** *ip-address*	No	—	Next cooperative (COOP) message
Disable redundancy	[**no**] **redundancy**	No	—	Must immediately configure on other key servers

When a timeout is caused by a pseudotime synchronization, the key server checks if either the KEK or the TEK timer is scheduled to expire in next 60 seconds, and if so, combines that timeout with the pseudotime

synchronization timeout. That is, the rekey acts as both a TEK or KEK rekey and a pseudotime synchronization timeout rekey. See the "Time-Based Antireplay" section for more information on pseudotime synchronization.

Retransmitting a Rekey

Multicast rekeys are retransmitted by default. For unicast rekeys, if the key server does not receive the ACK, it retransmits the rekey. In either case, before retransmitting a rekey, the key server checks if there is a TEK or KEK rekey scheduled in the next 120 seconds. If so, it stops the current retransmission and waits for the scheduled rekey to happen.

Group Member Access Control List

For GET VPN, the traffic that has to be protected is defined statically on the key server using the ACL. The group member gets information about what has to be protected from the key server. This structure allows the key server to choose and change the policy dynamically as needed. In Secure Multicast, the key server ACL is defined inclusively. The ACL includes only the exact traffic that should be encrypted, with an implicit deny causing all other traffic to be allowed in the clear (that is, if there is no permit, all other traffic is allowed).

GET VPN employs a different philosophy: The definition of which packets should be encrypted is delivered independently. GET VPN supports only statically defined traffic selectors. Policy can be defined by using both deny and permit ACLs on the key server. Only the deny ACL is allowed to be manually configured on a group member. The policies that are downloaded from the key server and configured on the group member are merged. Any ACL that is configured on the group member has predominance over what is downloaded from the key server.

After the group member gets the ACL from the key server, the group member creates a temporary ACL and inserts it into the database. This ACL will be deleted if the group member is removed from the GDOI group for any reason. The packets that are going out of the interface are dropped by the group member if a packet matches the ACL but no IPsec SA exists for that packet.

The key server can send a set of traffic selectors, which may not exactly match the group member ACL on the group member. If such differences occur, the differences have to be merged and resolved. Because the group member is more aware of its topology than the key server, the downloaded ACLs are appended to the group member ACL. The group member ACL (except the implicit deny) is inserted into the database first, followed by the downloaded key server ACL. The database is prioritized, and the database search stops whenever a matched entry is found.

For information about configuring a group member ACL, see the "Configuring Group Member ACLs" section.

Behavior of a Group Member When Security Policy Changes

The behavior of a group member changes when ACL changes or any other policy changes are made in the key server. The effect of different policy changes on the behavior of the group members is explained in the following three scenarios.

Scenario 1

In the following example, the ACL has been initially configured to permit host A and host B.

```
ip access-list extended get-acl
permit ip host A host B
permit ip host B host A
```

Then the ACL is changed to permit host C and host D in the key server:

```
ip access-list extended get-acl
permit ip host C host D
permit ip host D host C
```

ACL changes affect the behavior of the group member in the following ways:

- Key server sends out a rekey to all group members immediately.
- Group member sends traffic between host A and host B in clear text immediately after rekey.
- Group member sends traffic between host C and host D in encrypted text immediately after rekey.

Note GETVPN group members of Cisco ASR 1000 Series Aggregation Services Routers and Cisco ISR G2 routers behave differently after a rekey (either triggered or periodic) that follows a ACL change or any other policy change in the key server. The group members of Cisco ISR G2 routers install the new policy without a full reregistration, while the group members of Cisco ASR 1000 Series Aggregation Services Routers will reregister to get the updated policy.

Scenario 2

The behavior of a group member changes when policy updates and transform set and time-based antireplay (TBAR) changes are made to the key server.

In this scenario, it is assumed that:

- The transform set has been changed from ESP-3DES to ESP-AES.
- The policy change occurs at 1000 seconds before the current TEK lifetime expires.

These policy changes affect the behavior of the group member in the following ways:

- The key server sends out a rekey of both old SAs (3DES) and new SAs (AES).
- Group member continues to use the old SA (3DES) for 1000 seconds until it expires.
- After the old SA expires, the group member automatically switches over to new SAs (AES).

Scenario 3

The behavior of a group member changes when other policy updates in the key server involve both ACL changes and other changes like the transform set or TBAR.

In this scenario it is assumed that:

- The ACL has been updated as specified in Scenario 1.
- The transform set was changed from ESP-3DES to ESP-AES.
- The policy change occurs 1000 seconds before the current TEK lifetime expires.

ACL changes and other policy updates affect the behavior of the group member in the following ways:

- The key server sends out a rekey that consists of both old SAs (3DES) and new SAs (AES).
- The group member sends traffic between host A and host B in clear text immediately after rekey.

- The group member sends encrypted traffic between host C and host D using old SAs (3DES) for 1000 seconds until its TEK lifetime expires.
- When old SAs (3DES) expire, the group member automatically switches to new SAs to encrypt traffic between host C and host D in AES.

Time-Based Antireplay

Antireplay is an important feature in a data encryption protocol such as IPSec (RFC 2401). Antireplay prevents a third party from eavesdropping on an IPsec conversation, stealing packets, and injecting those packets into a session at a later time. The time-based antireplay mechanism helps ensure that invalid packets are discarded by detecting the replayed packets that have already arrived at an earlier time.

GET VPN uses the Synchronous Antireplay (SAR) mechanism to provide antireplay protection for multisender traffic. SAR is independent of real-world Network Time Protocol (NTP) clock or sequential-counter mechanisms (which guarantee packets are received and processed in order). A SAR clock advances regularly. The time tracked by this clock is called pseudotime. The pseudotime is maintained on the key server and is sent periodically to the group members within a rekey message as a time-stamp field called pseudoTimeStamp. GET VPN uses a Cisco proprietary protocol called Metadata to encapsulate the pseudoTimeStamp. Group members have to be resynchronized to the pseudotime of the key server periodically. The pseudotime of the key server starts ticking from when the first group member registers. Initially, the key server sends the current pseudotime value of the key server and window size to group members during the registration process. New attributes, such as time-based replay-enabled information, window size, and the pseudotime of the key server, is sent under the SA payload (TEK).

The group members use the pseudotime to prevent replay as follows: the pseudoTimeStamp contains the pseudotime value at which a sender created a packet. A receiver compares the pseudotime value of senders with its own pseudotime value to determine whether a packet is a replayed packet. The receiver uses a time-based antireplay "window" to accept packets that contain a time-stamp value within that window. The window size is configured on the key server and is sent to all group members.

Note You should not configure time-based antireplay if you are using a Cisco VSA as a group member.

The figure below illustrates an antireplay window in which the value PTr denotes the local pseudotime of the receiver, and W is the window size.

Figure 121: Antireplay Window

Clock Synchronization

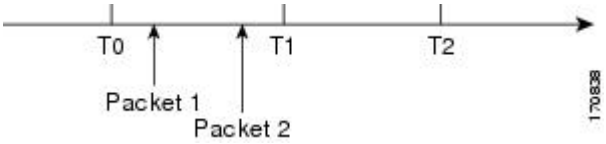
Clocks of the group members can slip and lose synchronization with the key server. To keep the clocks synchronized, a rekey message (multicast or unicast, as appropriate), including the current pseudotime value of the key server, is sent periodically, either in a rekey message or at a minimum of every 30 minutes to the group member. If a packet fails this antireplay check, the pseudotime of both the sender and receiver is printed, an error message is generated, and a count is increased.

To display antireplay statistics, use the **show crypto gdoi group** *group-name* **gm replay** command on both the sender and receiver devices. If the configuration is changed by the administrator to affect the replay method of the size configuration, the key server initiates a rekey message.

Interval Duration

A tick is the interval duration of the SAR clock. Packets sent in this duration have the same pseudoTimeStamp. The tick is also downloaded to group members, along with the pseudotime from the key server. For example, as shown in the figure below, packets sent between T0 and T1 would have the same pseudoTimeStamp T0. SAR provides loose antireplay protection. The replayed packets are accepted if they are replayed during the window. The default window size is 100 seconds. It is recommended that you keep the window size small to minimize packet replay.

Figure 122: SAR Clock Interval Duration



Antireplay Configurations

The Antireplay feature can be enabled under IPsec SA on a key server by using the following commands:

- **replay time window-size**—Enables the replay time option, which supports the nonsequential, or time-based, mode. The window size is in seconds. Use this mode only if you have more than two group members in a group.
- **replay counter window-size**—Enables sequential mode. This mode is useful if only two group members are in a group.
- **no replay counter window-size**—Disables antireplay.

Control-Plane Time-Based Antireplay

Rekey Pseudotime Check

The rekey pseudotime check between key servers and group members is conducted as follows:

- The group member calculates the allowable pseudotime difference between the key server and its own as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.
- The group member accepts any rekey with a pseudotime larger than its own and updates its own pseudotime to the larger value. If the difference is larger than the calculated allowable pseudotime difference, it also generates the following syslog message:

```
*Jul 28 22:56:37.503: %GDOI-3-PSEUDO_TIME_LARGE: Pseudotime difference between key server
(20008 sec) and GM (10057 sec) is larger than expected in group GET. Adjust to new pseudotime
```

- If the group member receives a rekey with a pseudotime smaller than its own but within the allowable difference, the group member accepts the rekey and updates its pseudotime value to the rekey pseudotime value.

- If the group member receives a rekey with a pseudotime smaller than its own but exceeding the allowable difference, the group member drops the rekey message and generates the following syslog message:

```
*Jul 28 23:37:59.699: %GDOI-3-PSEUDO_TIME_TOO_OLD: Rekey received in group GET is too old
and fail PST check: my_pst is 22490 sec, peer_pst is 10026 sec, allowable_skew is 30 sec
```

ANN Message Pseudotime Handling in the Secondary Key Server

Cooperative key server announcement (ANN) messages are used to synchronize policy and group-member information between cooperative key servers.

The secondary key server handles ANN messages as follows:.

- The secondary key server calculates the allowable ANN message pseudotime as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.
- If the secondary key server receives an ANN message from the primary key server with a larger pseudotime, it does the following:
- It updates its pseudotime to the primary key server's value.
- If the pseudotime difference is larger than allowable, it generates the following syslog message:

```
*Jul 28 23:48:56.871: %GDOI-4-GDOI_ANN_TIMESTAMP_LARGE: COOP_KS ANN received from KS 10.0.8.1
 in group GET has pseudotime bigger than myself. Adjust to new pseudotime:
my_old_pst is 23147 sec, peer_pst is 30005 sec
```

- If the secondary key server receives an ANN message from the primary key server with a smaller pseudotime, it behaves as follows:
  - If the difference is within the allowable range, the secondary key server accepts it and updates its pseudotime to the primary key server's value.
  - If the difference exceeds the allowable range, it generates the following syslog message:

```
*Jul 28 23:42:12.603: %GDOI-4-GDOI_ANN_TIMESTAMP_TOO_OLD: COOP_KS ANN from KS 10.0.8.1 in
group GET is too old and fail PST check:
my_pst is 22743 sec, peer_pst is 103 sec, allowable_skew is 10 sec
```

If, after three retransmit requests, the secondary key server has still not received any ANN message with a valid pseudotime, it starts blocking new group-member registrations, as follows:

```
*Jul 28 23:38:57.859: %GDOI-5-COOP_KS_VALID_ANN_TIMER_EXPIRED: This sec-KS has NOT received
 an ANN with valid pseudotime for an extended period in group GET. It will block new group
 members registration temporarily until a valid ANN is received
*Jul 29 00:08:47.775: %GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER: This key server temporarily
blocks group member with ip-addr 10.0.0.2 from registering in group GET as it has not
received an ANN with valid pseudotime for prolonged period
```

The secondary key server resumes its group member registration functionality if any of the following happens:

- It receives an ANN with a valid pseudotime from the primary key server.
- It becomes a primary key server itself.
- The **clear crypto gdoi group** command is executed on the secondary key server.

ANN Message Pseudotime Handling in the Primary Key Server

The primary key server handles ANN messages as follows:

- It calculates the allowable ANN message pseudotime as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.
- It accepts from the secondary key server ANN messages that have a smaller pseudotime but are within the allowable difference.
- It rejects ANN messages that have a smaller pseudotime but exceed the allowable difference.

During a network merge, the following conditions apply:

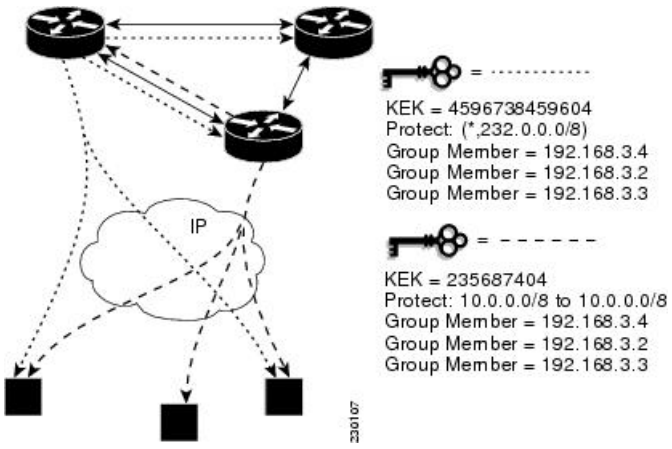
- The new primary key server always picks the larger pseudotime between the two key servers.
- If the difference is larger than the calculated allowable pseudotime difference, the new primary key server sends out rekeys to all group members to update their pseudotime. It also generates the following syslog messages:

```
*Jul 28 23:42:41.311: %GDOI-5-COOP_KS_ELECTION: KS entering election mode in group GET
(Previous Primary = NONE)
*Jul 28 23:42:41.311: %GDOI-4-GDOI_ANN_TIMESTAMP_LARGE: COOP_KS ANN received from KS 10.0.9.1
 in group GET has PST bigger than myself. Adjust to new pseudotime:
my_old_pst is 0 sec, peer_pst is 22772 sec
*Jul 28 23:43:16.335: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 10.0.8.1 in group GET transitioned
to Primary (Previous Primary = NONE)
*Jul 28 23:43:16.347: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group GET
from address 10.0.8.1 with seq # 1
```

Cooperative Key Server

The figure below illustrates cooperative key server key distribution. The text following the illustration explains the Cooperative Key Server feature.

Figure 123: Cooperative Key Server Key Distribution



Cooperative key servers provide redundancy to GET VPN. Multiple key servers are supported by GET VPN to ensure redundancy, high availability, and fast recovery if the primary key server fails. Cooperating GDOI key servers jointly manage the GDOI registrations for the group. Each key server is an active key server, handling GDOI registration requests from group members. Because the key servers are cooperating, each key

server distributes the same state to the group members that register with it. Load balancing is achieved because each of the GDOI key servers can service a portion of the GDOI registrations.

The primary key server is responsible for creating and distributing group policy. When cooperative key server key distribution occurs, one key server declares itself as primary, creates a policy, and sends the policy to the other secondary key server. The secondary key server declares the primary key server as primary key server when it gets the policy and ends the election mode. The secondary key server now also blocks GM registration while the cooperative key server key distribution is in progress. This change allows the cooperative key server distribution to become more efficient because it saves time. For example, the syslog warning message similar to the following is displayed during distribution:

```
00:00:16: %GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER_ELECTION: This KS temporarily blocks GM
with ip-addr 10.0.4.1 from registering in group diffint as the KS election is underway
```

The primary key server periodically sends out (or broadcasts) group information updates to all other key servers to keep those servers in synchronization. If the secondary key servers somehow miss the updates, they contact the primary key server to directly request information updates. The secondary key servers mark the primary key server as unreachable (that is, “dead”) if the updates are not received for an extended period.

When a new policy is created on a primary key server, regardless of which key server a group member may be registered with, it is the responsibility of the primary key server to distribute rekey messages to GDOI group members.

In a cooperative-key-server setting, the rekey sequence number is synchronized between the primary and secondary key servers.

In a network merge, the key servers pick the larger of the rekey sequence numbers that they have between them.

If you are supporting more than 300 group members in your cooperative key server setup, you should increase the buffer size by using the **buffers huge size** command.

If the registration interface that is used in the GETVPN Group Configurtion on the key server is shutdown, a network spit will occur. If the interface is not the forwarding interface, as in the loopback interface which is the recommended configuration, rekeys will still be sent to the GMs from all of the KSs in the group. You cannot turn off the keyservers by shutting down the interface. To safely turn off the keyservers, use the **no crypto gdoi group** *group name* command.

The following example shows the registration interface that is referenced in the GETVPN Group Configurtion on the key server.

```
crypto gdoi group groupA
identity number 111
server local
  sa ipsec 10
   profile groupA
   match address ipv4 groupA-crypto-policy
   no replay
  no tag
  address ipv4 a.b.c.d
  redundancy
   local priority 250
   peer address ipv4 a.b.c.d
   peer address ipv4 a.b.c.d
```

Announcement Messages

Announcement messages are secured by IKE Phase 1 and are sent as IKE notify messages. Authentication and confidentiality that are provided by IKE is used to secure the messaging between the key servers. Antireplay

protection is provided by the sequence numbers in the announcement messages. Announcement messages are periodically sent from primary to secondary key servers.

Announcement messages include the components, described in the following sections that help maintain the current state.

Sender Priority of a Key Server

This value describes the priority of the sender, which is configurable using the CLI. The key server with the highest priority becomes the primary key server. If the priority values are the same, the key server with the highest IP address becomes the primary key server.

Maintaining the Role of the Sender

During the synchronization period, if the key servers are at geographically dispersed locations, they may suffer a network-partitioning event. If a network-partitioning event occurs, more than one key server can become the primary key server for a period of time. When the network is operating normally again and all the key servers find each other, they need to be told the current role of the sender so the key servers can attain their proper roles.

Request for a Return Packet Flag

All messages are defined as one-way messages. When needed, a key server can request the current state from a peer to find out its role or request the current state of the group.

Group Policies

The group policies are the policies that are maintained for a group, such as group member information and IPsec SAs and keys.

Antireplay functionalities and incorporated Cooperative announcement messages are supported. The primary key server updates the pseudotime value, sending it to all secondary key servers in the group. The secondary key servers should synchronize their SAR clocks to this updated value.

ANN Message Sequence Number Check Between Cooperative Key Servers

The following describes the sequence number check between cooperative key servers:

- Cooperative key servers drop any ANN message with a sequence number smaller than or equal to that of the last received ANN message.
- The ANN message is accepted if the sequence number is larger than that of the last received rekey message, no matter how large the difference.
- If a key server is reloaded, a new IKE session is created between the peers, and the reloaded key server's ANN sequence number will start with zero. In this case, the other side will accept the ANN message with any sequence number.

Change Key Server Role

In a network of cooperative key servers, the primary server is elected based on its highest priority at the time of election. The other key servers have secondary status. If the primary key server is detected as being dead or if its role changes, the **clear crypto gdoi ks coop role** command allows you to reset the cooperative role of the primary key server.

If the **clear crypto gdoi ks coop role** command is executed on a secondary key server, the election is triggered on that secondary key server although that server would most likely remain a secondary key server because there has been an elected primary key server. However, if the **clear crypto gdoi ks coop role** command is executed on the primary key server, the primary key server is reassigned to a secondary role, and as a result, a new election that involves all the key servers is triggered. If the previous primary server has the highest priority (of all the key servers), it again becomes the primary server. If the previous primary server does not have the highest priority, the server having the highest priority is elected as the new primary server.

Receive Only SA

For multicast traffic using the GDOI protocol, bidirectional SAs are installed. The Receive Only feature enables an incremental deployment so that only a few sites can be verified before bringing up an entire network. To test the sites, one of the group members should send encrypted traffic to all the other group members and have them decrypt the traffic and forward the traffic "in the clear." Receive Only SA mode allows encryption in only the inbound direction for a period of time. (See the steps for the Receive Only SA process.) If you configure the **sa receive-only** command on the key server, Steps 2 and 3 happen automatically.

1. Mark IPsec SAs as "receive-only" on the GDOI key server.

This action allows the group members to install SAs in the inbound direction only. Receive-only SAs can be configured under a crypto group. (See the "Configuring the Group ID Server Type and SA Type" section.)

1. Mark GDOI TEK payloads as "receive only."

If the **sa receive-only** command is configured, all TEKs under this group are going to be marked "receive only" by the key server when they are sent to the group member.

1. Install one-way IPsec flows.

Every time a GDOI group member receives an IPsec SA from the key server that is marked as "receive only," the group member installs this IPsec SA only in the inbound direction rather than in both incoming and outgoing directions.

1. Test individual group members using the following local-conversion commands:
2. **crypto gdoi gm ipsec direction inbound optional**
3. **crypto gdoi gm ipsec direction both**

First, individually convert each of the group members to passive mode (this change tells the outbound check that there is a valid SA) and then to bidirectional mode.

1. Globally convert from "receive only" to "receive and send."

The following method can be used when the testing phase is over and "receive only" SAs have to be converted to bidirectional SAs.

Global Conversion

Remove the **sa receive-only** command under the group. Removing the **sa receive-only** command creates new IPsec SAs for this group and causes a rekey. On receipt, group members reinstall the SA in both directions and begin to use it in passive mode. Because the SA cannot remain in passive mode forever, the group members change those SAs to receive or send mode if there is no rekey in 5 minutes. The conversion from passive mode to bidirectional encryption mode is automatic and does not require the administrator to do anything.

Passive SA

The Passive SA feature allows you to configure a group member so that it is in passive mode permanently. By using the Passive SA feature, you will avoid having to use the **crypto gdoi gm ipsec direction inbound optional** privileged EXEC command, which is not persistent after a router reload and can be overridden by key server configuration from a rekey. Having the group member in passive mode benefits network testing and debugging during migration to GET VPN, and it provides complete encryption protection during the migration. The group-member passive-mode configuration has higher priority over a key server configuration. The **crypto gdoi gm ipsec direction inbound optional** privileged EXEC command can override the configuration until the next rekey, which will bring back the group member and key server configuration.

To configure the Passive SA feature, see the "Configuring Passive SA" section.

Enhanced Solutions Manageability

Several **show** and **debug** commands are supported to help verify functionality. See the "Activating Fail-Close Mode" section for details.

Support with VRF-Lite Interfaces

The VRF-Lite application supports segmentation of traffic in the control and forwarding planes by keeping the routing tables separate for each user group (or VPN) and forwarding the traffic on the associated or dedicated interfaces of each user group.

There are some deployment scenarios in which remote sites that are connecting to an MPLS VPN network might be extending segmentation from a campus to the WAN. In such an extended segmentation case, a CE-PE interface on a CE (group member or key server) device "bounds" to its associated virtual routing and forwarding (VRF) instance. This VRF interface connects to an MPLS PE device where it is directly mapped to its associated Border Gateway Protocol (BGP) VRF process, in which case the crypto map is applied to a VRF interface. No other configuration changes are necessary.

Authentication Policy for GM Registration

GMs can authenticate to the key server at registration time using preshared keys or Public Key Infrastructure (PKI). Preshared keys are easy to deploy but must be managed proactively. We recommend that you deploy a peer-based preshared key instead of defining a default key (the key defined with an address of 0.0.0.0) for all the devices in the network. Preshared keys should be updated regularly (every few months).

Note A preshared key can be updated on a key server-group member (KS-GM) peer basis without affecting the crypto data plane or control plane because rekeys are secured using the KEK. It is important to ensure that a GM can re-register to each ordered set of key servers using the newly assigned preshared key.

PKI uses its infrastructure to overcome the key management difficulties encountered when preshared keys are used. The PKI infrastructure acts as a certificate authority (CA) where router certificates are issued and maintained. However, using PKI during IKE authentication is computationally intensive. In PKI deployments, key server capacity, design, and placement become important.

For added security, GET VPN also supports GM authorization using either preshared keys or PKI. For more information, see the "GET VPN Authorization" section.

GET VPN GM Authorization

GET VPN GM authorization can be done using preshared keys or PKI. It is a best practice to turn on GET VPN authorization. When a key server serves multiple GDOI groups, key server authorization is required to prevent a GM in one group from requesting keys and policies from another group. The ISAKMP authorization confirms that the GM is allowed to request GDOI attributes from the key server while the GDOI authorization confirms that the GM is allowed to request GDOI attributes from a specific group configured in the key server.

GDOI authorization is based on the ISAKMP identity sent by a GM. If a GM sends an IP address as an identity, then only an authorization address is used for authorization. If a GM sends a distinguished name (DN) or hostname, then an authorization identity is used. Using an IP address as an identity will bypass authorization matching a DN or hostname and vice versa. To ensure that only GMs with a specific DN can connect (and no GMs using another identity can connect), you must specify **deny any** in the authorization address.

GM Authorization Using Preshared Keys

GET VPN supports GM authorization using the IP address when preshared keys are used. An ACL matching the WAN addresses (or subnets) of the GM can be defined and applied to the GET VPN group configuration. Any GM whose IP addresses match the ACL authorizes successfully and can register to the key server. If a GM IP address does not match the ACL, the key server rejects the GM registration request.

In cases of unsuccessful authorization, the following syslog message is generated:

```
%GDOI-1-UNAUTHORIZED_IPADDR: Group getvpn received registration from
unauthorized ip address: 10.1.1.9
```

GM Authorization Using PKI

GET VPN supports GM authorization using the commonly used DN or fully qualified domain name (FQDN) when PKI is used. The **authorization identity** command is used to activate GM authorization. A crypto identity matching certain fields in the GM certificate (typically—organizational unit [OU]) can be defined and applied to the GET VPN group configuration. Use the **crypto identity** command to define a crypto identity.

Any GM whose certificate credentials match the ISAKMP identity is authorized and can register to the key server. For example, if all GM certificates are issued with OU=GETVPN, a key server can be configured to check (authorize) that all GMs present a certificate having OU=GETVPN. If the OU in the certificate presented by a GM is set to something else, the GM will not be authorized to register to the key server.

If authorization is unsuccessful, the following syslog message is generated:

```
%GDOI-1-UNAUTHORIZED_IDENTITY: Group getvpn received registration from
unauthorized identity: Dist.name: hostname=GroupMember-1, ou=TEST
```

Rekey Functionality in Protocol Independent Multicast-Sparse Mode

Multicast rekeying can be used with all modes of multicast. The **rekey retransmit** command should be used whenever the Protocol Independent Multicast-sparse mode (PIM-SM) is configured because the PIM-SM shortest path tree (SPT) can be torn down if it does not receive continuing traffic. When traffic resumes, PIM-SM must reestablish the SPT. Retransmitting rekey packets increases the chance that group members receive the rekeys when PIM-SM is setting up the SPT.

Fail-Close Mode

Until a group member registers with a key server, traffic passing through the group member is not encrypted. This state is called "fail open." To prevent unencrypted traffic from passing through a group member before that member is registered, you can configure the Fail-Close feature. If the feature is configured, an implicit "permit ip any any" policy is installed, and all unencrypted traffic passing through the group member is dropped (this state is called fail-close mode).

The fail-close function can also be achieved by configuring an interface ACL. However, the Fail-Close feature is more manageable and is easier to implement than ACL lists.

If you configure the Fail-Close feature, you can still allow specific unencrypted traffic to pass through the group member by configuring the **match address** command (**match address** {*access-list-number* | *access-list-name*}). This explicit "deny" ACL is added before the implicit "permit ip any any" to allow denied (unencrypted) traffic to pass through the group member.

After the group member has successfully completed its registration, the fail-close policy, both explicit and implicit, is removed, and the group member behaves as it did before the Fail-Close feature was configured.

Guidelines for Using the Fail-Close Feature

When you are configuring a crypto map to work in fail-close mode, you must be careful. If the fail-close ACL is defined improperly, you may lock yourself out of the router. For example, if you use Secure Shell (SSH) to log in to the router through the interface with the crypto map applied, you have to include the **deny tcp any eq** *port* **host** *address* command line under the fail-close ACL. You may also need to include the routing protocol that the router is using (such as **deny ospf any any**) to find the path to the key server. It is suggested that you configure fail-close and its ACL first, and then verify the fail-close ACL using the **show crypto map gdoi fail-close** *map-name* command. After you have checked your fail-close ACL and are confident that it is correct, you can make the crypto map work in the fail-close mode by configuring the **activate** command. Fail-close is not activated until you have configured the **activate** command.

The fail-close ACL is configured from the group-member perspective. The fail-close ACL is configured on group member as follows:

access-list 125 deny ip host *host1-ip-addr host2-ip-addr*

In fail-close mode, all IP traffic from host1 to host2 will be sent out by group member 1 in clear text. In addition, the inbound mirrored traffic (that is, IP traffic from host2 to host1) is also accepted by GM1 in clear text.

Note All IP traffic matching deny entries are sent out by the group member in clear text.

The inbound traffic is matched to the mirrored access list.

The fail-close access list follows the same rules as the group member access list. For more information, see the "Group Member Access Control List" section.

You need not configure the **deny udp any eq 848 any eq 848** command to make the GDOI registration go through. The code itself checks whether it is a GDOI packet for a particular group member from the key server to which it is configured. If it is a GDOI packet for this group member, the packet is processed. But for a scenario in which the key server is behind group member 1, if group member 1 cannot register successfully with the key server, other group members also will not be able to register unless an explicit **deny udp any eq 848 any eq 848** command line is configured for group member 1. However, if the Fail-Close feature is properly configured, even if a group member fails to register with a key server, you will be able to ensure that

no unwanted traffic can go out “in the clear.” But you can allow specified traffic to go out in the clear, in which case registration packets from other group members will be able to reach the key server through group member 1 even if it fails to get registered.

For information on configuring fail-close mode, see the "Activating Fail-Close Mode" section.

To verify whether fail-close mode is activated, use the **show crypto map gdoi fail-close** command.

Fail-close Revert

In the fail close mode, before registering in the fail close mode, group member applies its local fail close policy and manages the traffic accordingly. After registration, group member applies the policy downloaded from key server and handles traffic accordingly.

When there is no rekey or the group member is not able to re-register to the key server, the group member uses the same downloaded policy from the key server. It leads to packet drop as there is no key for encryption or decryption. Fail-close Revert enables group members to return to the fail close mode and to remove the downloaded key server policy. This happens only if fail-close revert is enabled on the group member.

This fail-close revert triggers when all active SAs expire and all the key servers are not reachable for re-registration. Clearing the IPsec SAs manually by using the command “clear crypto sa” does not provide the intended behavior of the feature. However “clear crypto gdoi” will revert to fail close mode in case of a key server unreachability.

To know about feature configuration steps, see the section, “Configuring Fail Close Revert”.

Create MIB Object to Track a Successful GDOI Registration

The routing plane and crypto plane for GET VPN must be synchronized to avoid null routes. A GET VPN null route occurs during the following situations:

- GMs fail to register to a KS that has no active TEKs to encrypt or decrypt traffic.
- GMs TEK SAs have expired but do not receive new keys from KS through rekey or reregistration.
- GMs receive rekeys from KS, but errors occur when installing SAs to a crypto engine.

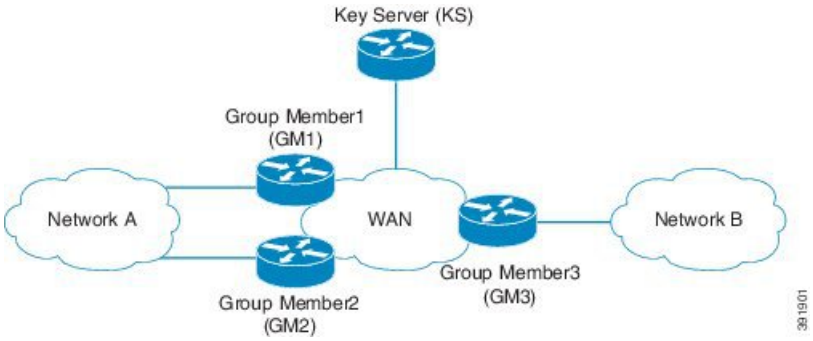
The Create MIB Object to Track a Successful GDOI Registration feature introduces a new MIB object in the GDOI MIB to indicate the number of active TEKs in a group.

GET VPN Routing Awareness for BGP

The routing plane and crypto plane for GET VPN must synchronize to avoid null routes. When a group member (GM) successfully registers to a key server (KS), no security policies or keys are installed on the GM. However, the GM may still advertise the routes of its protected network to other GMs.

The following diagram explains the creation of a null route.

Figure 124: Null Route Creation



1. Group Member1, Group Member2, Group Member3 boot up and establish a routing adjacency with the WAN.
2. Group Member1 and Group Member2 advertises the prefix for Network A into the WAN. The preferred path for traffic from Network B to Network A is through Group Member1.
3. Group Member3 advertises Network B into the WAN. The preferred path for traffic Network A to Network B is through Group Member1
4. KS defines the security to protect all traffic between Network A and Network B
5. Group Member1 and Group Member3 (as well as Group Member2) successfully obtain security keys from KS and protect all traffic between Network A <-> Network B.
6. Group Member1 fails to receive updated keys or policy and fails to reregister to a KS while Group Member2 and Group Member3 successfully obtain keys.
7. Routing protocols continue to prefer the path through Group Member1 for all Network A between Network B traffic.
8. Group Member1 drops all traffic flowing between Network A and Network B because the policy/keys are invalid.

When the host in Network B sends traffic to a host in Network A, the traffic will be encrypted by Group Member3 and sent to Network A via Group Member1 (the preferred path). However, Group Member1 will drop the packet because it has no policy or current keys to decrypt traffic. As a result, the traffic is dropped and a null route is created. Likewise, when a host in Network A sends traffic to a host in Network B, the traffic will be directed to Group Member1 (the preferred path) and dropped due to lack of policy or current keys in Group Member1. The appropriate behavior is for the traffic to be diverted and rerouted via Group Member2 while Group Member1 has no policy or keys.

The GET VPN Routing Awareness for BGP feature prevents routing absence by tracking the GETVPN GM crypto state and by applying the tracking information to perform bidirectional conditional route filtering on the GM.

Bidirectional Conditional Route Filtering

The bidirectional conditional route filtering supports different routing protocols, such as BGP, OSPF, EIGRP, RIPv2, etc. The EOT tracks the GET VPN GM crypto status and conditionally enables or disables specific route-map entries based on the EOT value. The following is a sample configuration to monitor the GET VPN GM crypto state.

```
route-map bgp-policy-out permit 10
 match ip address register-int-Only
route-map bgp-policy-out permit 20
 match track 99
 match ip address orig_route_map_acl_out
route-map bgp-policy-out deny 30

route-map bgp-policy-in permit 10
 match ip address noc
route-map bgp-policy-in permit 20
 match track 99
 match ip address orig_route_map_acl_in
route-map bgp-policy-in deny 30

ip access-list standard noc
 permit 1.1.1.0                                  <---- NOC subnet with Keyserver (KS)
ip access-list standard register-int-Only
 permit 2.2.2.2                                  <---- registration interface ip of the
GM itself
ip access-list standard orig_route_map_acl_in       <---- original inbound route-map ACL
 permit a.b.c.d
 permit .......
ip access-list standard orig_route_map_acl_out      <---- original outbound route-map ACL
 permit e.f.g.h
 permit .......

router bgp 64600
 no synchronization
 bgp router-id xxxxxxxx
 bgp log-neighbor-changes
 network xxxxxxxxxx mask 255.255.255.255
 network xxxxxxxxxx mask 255.255.255.252
 neighbor xxxxxxxxxx remote-as 65000
 neighbor xxxxxxxxxx description PE
 neighbor xxxxxxxxxx route-map bgp-policy-in in
 neighbor xxxxxxxxxx route-map bgp-policy-out out
```

In the above example, the **match track 99** command is specified to monitor the GET VPN GM crypto state. If GM works properly, the **match track 99** command returns a value *true* and the GM advertises or receives the following routes:

- Outbound—The routes to reach the GM registration interface and the routes permitted by inbound route map access control list (ACL) “orig_route_map_acl_out.”
- Inbound—The routes to reach the NOC and the routes permitted by outbound route map ACL “orig_route_map_acl_in” received from peers.

On the other hand, if GM does not work properly, the **match track 99** command returns a value *false* and the GM advertises or receives the following routes only:

- Outbound—The routes to reach the GM registration interface.
- Inbound—The routes to reach the NOC subnet.

Cisco Group Encrypted Transport VPN System Logging Messages

The table below lists GET VPN system logging (also called syslog) messages and explanations.

Table 288: GET VPN System Logging Messages

Message	Explanation
COOP_CONFIG_MISMATCH	The configuration between the primary KS and secondary KS are mismatched.
COOP_KS_ADD	A KS has been added to the list of cooperative KSs in a group.
COOP_KS_ELECTION	The local KS has entered the election process in a group.
COOP_KS_REACH	The reachability between the configured cooperative KSs is restored.
COOP_KS_REMOVE	A KS has been removed from the list of cooperative KSs in a group.
COOP_KS_TRANS_TO_PRI	The local KS transitioned to a primary role from being a secondary server in a group.
COOP_KS_UNAUTH	An unauthorized remote server tried to contact the local KS in a group. Could be considered a hostile event.
COOP_KS_UNREACH	The reachability between the configured cooperative KSs is lost. Could be considered a hostile event.
COOP_KS_VER_MISMATCH	KSs are running different versions of the Cisco IOS code.
COOP_PACKET_DROPPED	A hard limit set on the driver buffer size prevents the sending of packets this size or larger.
GDOI-3-GDOI_REKEY_SEQ_FAILURE	The rekey message is rejected because the sequence number antireplay check failed.
GDOI-3-GM_NO_CRYPTO_ENGINE	No crypto engine is found due to a lack of resources or an unsupported feature requested.
GDOI-3-PSEUDO_TIME_LARGE	The rekey has a larger pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-3-PSEUDO_TIME_TOO_OLD	The rekey has a smaller pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-4-GDOI_ANN_TIMESTAMP_ LARGE	The secondary KS receives from the primary KS an ANN that has a larger pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-4-GDOI_ANN_TIMESTAMP_ TOO_OLD	The secondary KS receives from the primary KS an ANN that has a smaller pseudotime that exceeds the calculated allowable pseudotime difference.

Message	Explanation
GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER	The secondary KS temporarily blocks a GM from registering in a group because it has not received a valid pseudotime from the primary KS.
GDOI-5-COOP_KS_VALID_ANN_ TIMER_EXPIRED	The secondary KS keeps receiving ANNs with invalid pseudotimes after three retransmits. The secondary KS temporarily blocks new group-member registration until a valid ANN is received.
GDOI_ACL_NUM	The ACL has too many entries. GDOI will honor only the first 100 ACL entries specified.
GDOI_REKEY_FAILURE	During GDOI rekey the payload parsing failed on this GM from the KS.
GM_ACL_MERGE	The ACL differences between a GM and KS are resolved and a merge took place.
GM_ACL_PERMIT	The GM can support only an ACL for “deny.” Any traffic matching the “permit” entry will be dropped.
GM_CLEAR_REGISTER	The **clear crypto gdoi** command has been executed by the local GM.
GM_CM_ATTACH	A crypto map has been attached for the local GM.
GM_CM_DETACH	A crypto map has been detached for the local GM.
GM_CONV_SA_DUPLEX	IPsec SAs have been converted to bidirectional mode in a group on a GM.
GM_CONV_SA_DUPLEX_LOCAL	IPsec SAs have been converted to bidirectional mode in a group on a GM by a CLI command.
GM_DELETE	A GM has been deleted in a group from a KS.
GM_ENABLE_GDOI_CM	A GM has enabled ACL on a GDOI crypto map in a group with a KS.
GM_HASH_FAIL	During GDOI registration protocol, a message sent by the KS has bad or no hash.
GM_INCOMPLETE_CFG	Registration cannot be completed because the GDOI group configuration may be missing the group ID, server ID, or both.
GM_NO_IPSEC_FLOWS	The hardware limitation for IPsec flow limit reached. Cannot create any more IPsec SAs.
GM_RE_REGISTER	The IPsec SA created for one group may have been expired or cleared. Need to re-register to the KS.

Message	Explanation
GM_RECV_DELETE	A message sent by the KS to delete the GM has been received.
GM_RECV_REKEY	Rekey received.
GM_REGS_COMPL	Registration complete.
GM_REJECTING_SA_PAYLOAD	During GDOI registration protocol, a proposal sent by the KS was refused by the local GM.
GM_REKEY_NOT_RECD	A GM has not received a rekey message from a KS in a group. Currently unimplemented.
GM_REKEY_TRANS_2_MULTI	A GM has transitioned from using a unicast rekey mechanism to using a multicast mechanism.
GM_REKEY_TRANS_2_UNI	A GM has transitioned from using a multicast rekey mechanism to using a unicast mechanism.
GM_SA_INGRESS	A received-only ACL has been received by a GM from a KS in a group.
GM_UNREGISTER	A GM has left the group.
KS_BAD_ID	A configuration mismatch exists between a local KS and a GM during GDOI registration protocol.
KS_BLACKHOLE_ACK	A KS has reached a condition of null route messages from a GM. Could be considered a hostile event.
KS_CLEAR_REGISTER	The **clear crypto gdoi** command has been executed by the local KS.
KS_CONV_SAS_DUPLEX	IPsec SAs have been converted to bidirectional mode in a group.
KS_CONV_SAS_INGRESS	IPsec SAs have been converted to receive-only mode in a group.
KS_FIRST_GM, GDOI, LOG_INFO	A local KS has received the first GM joining the group.
KS_GM_REJECTS_SA_PAYLOAD	During GDOI registration protocol, a proposal sent by the KS was refused by the GM.
KS_GM_REVOKED	During rekey protocol, an unauthorized member tried to join a group. Could be considered a hostile event.
KS_GROUP_ADD	A configuration command has been executed to add a KS in a group.

Message	Explanation
KS_GROUP_DELETE	A configuration command has been executed to remove a KS from a group.
KS_HASH_FAIL	During GDOI registration protocol, a message sent by the GM has a bad or no hash.
KS_LAST_GM	The last GM has left the group on the local KS.
KS_NACK_GM_EJECT	The KS has reached a condition of not receiving an ACK message from a GM and has been ejected.
KS_NO_RSA_KEYS	RSA keys were not created or they are missing.
KS_REGS_COMPL	The KS has successfully completed a registration in a group.
KS_REKEY_TRANS_2_MULTI	The group has transitioned from using a unicast rekey mechanism to a multicast mechanism.
KS_REKEY_TRANS_2_UNI	The group has transitioned from using a multicast rekey mechanism to using a unicast mechanism.
KS_SEND_MCAST_REKEY	Sending multicast rekey.
KS_SEND_UNICAST_REKEY	Sending unicast rekey.
KS_UNAUTHORIZED	During GDOI registration protocol, an unauthorized member tried to join a group. Could be considered a hostile event.
KS_UNSOL_ACK	The KS has received an unsolicited ACK message from a past GM or is under a DOS attack. Could be considered a hostile event.
PSEUDO_TIME_LARGE	A GM has received a pseudotime with a value that is largely different from its own pseudotime.
REPLAY_FAILED	A GM or KS has failed an antireplay check.
UNAUTHORIZED_IDENTITY	The registration request was dropped because the requesting device was not authorized to join the group.
UNAUTHORIZED_IPADDR	The registration request was dropped because the requesting device was not authorized to join the group.
UNEXPECTED_SIGKEY	An unexpected signature key was found that frees the signature key.
UNREGISTERED_INTERFACE	Receiving registration from unregistered interface. Stop processing it.

Message	Explanation
UNSUPPORTED_TEK_PROTO	Unexpected TEK protocol.

How to Configure Cisco Group Encrypted Transport VPN

Configuring a Key Server

Prerequisites

Before creating the GDOI group, you must first configure IKE and the IPsec transform set, and you must create an IPsec profile. For information about how to configure IKE and the IPsec transform set and to create an IPsec profile, see the "Related Documents" subsection of the "Additional References" section.

Configuring RSA Keys to Sign Rekey Messages

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

To configure RSA keys that will be used to sign rekey messages, perform the following steps. Omit this subtask if rekey is not in use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa general-keys label** *name-of-key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa general-keys label** *name-of-key*	Generates RSA keys that will be used to sign rekey messages. You are prompted to confirm the length (in bits)

	Command or Action	Purpose
	Example: `Router(config)# crypto key generate rsa general-keys label mykeys`	of the keys to be generated. Length of less than 2048 is not recommended.

What to Do Next

Configure the group ID, server type, and SA type. (See the "Configuring the Group ID Server Type and SA Type" section.)

Configuring the Group ID Server Type and SA Type

For a large number of sites, it is better to take precautions and add functionality incrementally, especially when migrating from any other encryption solutions like Dual Multipoint VPN (DMVPN). For example, instead of setting up all the CPE devices to encrypt the traffic bidirectionally, it is possible to configure one-way encryption so that only one or fewer members of a group are allowed to send encrypted traffic. Others are allowed to receive only encrypted traffic. After the one-way encryption is validated for one or a few members, bidirectional encryption can be turned on for all the members. This "inbound only" traffic can be controlled using the **sa receive only** command under a crypto group.

To configure the group ID, server type, and SA type, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **sa receive-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**sa receive-only** **Example:** `Router(config-local-server)# sa receive-only`	Specifies that an IPsec SA is to be installed by a group member as "inbound only."

What to Do Next

Remove the receive-only configuration on the key server so that the group members are now operating in bidirectional receive and send mode.

Configuring the Rekey

This section includes the following optional tasks:

Rekey is used in the control plane by the key server to periodically refresh the policy and IPsec SAs of the group. On the group-member side, instead of fully re-registering when timers expire for any other reasons, refreshing the registration with a rekey is more efficient. The initial registration is always a unicast registration.

The key server can be configured to send rekeys in unicast or multicast mode. The rekey transport mode is determined by whether the key server can use IP multicast to distribute the rekeys. If multicast capability is not present within the network of the customer, the key server will have to be configured to send rekeys using unicast messages.

Additional options for rekey use the **rekey authentication**, **rekey retransmit**, and **rekey address ipv4** commands. If unicast transport mode is configured, the **source address** command will have to be included to specify the source address of this unicast rekey message.

Multicast is the default transport type for rekey messages. The following bulleted items explain when to use rekey transport type multicast or unicast:

- If all members in a group are multicast capable, do not configure the **rekey transport unicast** command. The **no rekey transport unicast** command is not needed if the rekey transport type "unicast" was not configured previously under this group because multicast rekeys are on by default.
- If all members in a group are unicast, use the **rekey transport unicast** command.
- If you have mixed members in a group (that is, the majority are multicast, but a few are unicast), do not configure the **rekey transport unicast** command. The rekeys will be distributed using multicast to the majority of group members. The remainder of the group members that do not receive the multicast messages (unicast group members) will have to re-register to the key server when their policies expire. Mixed mode (that is, unicast and multicast rekey mode) is not supported.

If the **no rekey transport unicast** command is used, members in the GDOI group that are unable to receive the multicast rekey messages need to re-register with the key server to get the latest group policies. The re-registration forces the default transport type to multicast. If no transport type was configured previously, the multicast transport type will apply by default.

Prerequisites

Before configuring the **rekey authentication** command, you must have configured the router to have an RSA key generated using the **crypto key generate rsa** command and **general-keys** and **label** keywords (for example, "crypto key generate rsa general-key label my keys").

Configuring a Unicast Rekey

In the configuration task table, the address "ipv4 10.0.5.2" specifies the interface on the key server by which the unicast or multicast rekey messages are sent. This address is required for unicast rekeys, but it is optional for multicast rekeys. For multicast rekeys, the source address of the key server can be retrieved from the rekey ACL.

To configure a unicast rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **rekey transport unicast**
7. **rekey lifetime seconds** *number-of-seconds*
8. **rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions*
9. **rekey authentication mypubkey rsa** *key-name*
10. **address ipv4** *ipv4-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey transport unicast** **Example:** `Router(config-local-server)# rekey transport unicast`	Configures unicast delivery of rekey messages to group members.
Step 7	**rekey lifetime seconds** *number-of-seconds* **Example:** `Router(gdoi-local-server)# rekey lifetime seconds 300`	(Optional) Limits the number of seconds that any one encryption key should be used. • If this command is not configured, the default value of 86,400 seconds takes effect.
Step 8	**rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions* **Example:** `Router(gdoi-local-server)# rekey retransmit 10 number 3`	(Optional) Specifies the number of times the rekey message is retransmitted. • If this command is not configured, there will be no retransmits.

	Command or Action	Purpose
Step 9	**rekey authentication mypubkey rsa** *key-name* **Example:** `Router(gdoi-local-server)# rekey authentication mypubkey rsa mykeys`	(Optional) Specifies the keys to be used for a rekey to GDOI group members. • This command is optional if rekeys are not required. If rekeys are required, this command is required.
Step 10	**address ipv4** *ipv4-address* **Example:** `Router(gdoi-local-server)# address ipv4 209.165.200.225`	(Optional) Specifies the source information of the unicast rekey message. • If rekeys are not required, this command is optional. If rekeys are required, this command is required.

Configuring a Multicast Rekey

To configure a multicast rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **rekey address ipv4** {*access-list-name* | *access-list-number*}
7. **rekey lifetime seconds** *number-of-seconds*
8. **rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions*
9. **rekey authentication** {**mypubkey** | **pubkey**} **rsa** *key-name*
10. **exit**
11. **exit**
12. **access-list** *access-list-number* {**deny** | **permit**} **udp host** *source* [*operator*[*port*]] **host** *source* [*operator*[*port*]]
13. **interface** *type slot/ port*
14. **ip igmp join-group** *group-address* [**source** *source-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey address ipv4** {*access-list-name* | *access-list-number*} **Example:** `Router(gdoi-local-server)# rekey address ipv4 121`	Defines to which multicast subaddress range group members will register.
Step 7	**rekey lifetime seconds** *number-of-seconds* **Example:** `Router(gdoi-local-server)# rekey lifetime seconds 300`	(Optional) Limits the number of seconds that any one encryption key should be used. • If this command is not configured, the default value of 86,400 seconds takes effect.
Step 8	**rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions* **Example:** `Router(gdoi-local-server)# rekey retransmit 10 number 3`	(Optional) Specifies the number of times the rekey message is retransmitted. • If this command is not configured, there will be no retransmits.
Step 9	**rekey authentication** {**mypubkey** | **pubkey**} **rsa** *key-name*	(Optional) Specifies the keys to be used for a rekey to GDOI group members.

	Command or Action	Purpose
	Example: `Router(gdoi-local-server)# rekey authentication mypubkey rsa mykeys`	• This command is optional if rekeys are not required. If rekeys are required, this command is required.
Step 10	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI server local configuration mode.
Step 11	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode.
Step 12	**access-list** *access-list-number* {**deny** | **permit**} **udp host** *source* [*operator*[*port*]] **host** *source* [*operator*[*port*]] **Example:** `Router(config)# access-list 121 permit udp host 10.0.5.2 eq 848 host 239.0.1.2 eq 848`	Defines an extended IP access list.
Step 13	**interface** *type slot/ port* **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 14	**ip igmp join-group** *group-address* [**source** *source-address*] **Example:** `Router(config-if)# ip igmp join-group 232.2.2.2 source 10.1.1.1`	Configures an interface on the router to join the specified group or channel. **Note** Use this command to manually join the stream when the key server is not reachable via the same interface as the one configured with the crypto map.

Configuring Group Member ACLs

All IP traffic matching deny entries are sent out by the group member in clear text. The inbound traffic is matched to the mirrored access list.

Note The recommended method to add or delete an entry in the Group Member ACL is to first create a copy of the existing Group Member ACL with a different name and then add or delete the entry in this new ACL, after which, you should replace the existing group member ACL under the GDOI crypto map with the newly created Group Member ACL. If you do not follow this recommended method, it might lead to an unexpected behavior.

To configure group member ACLs, perform this task (note that a group member access list can contain only deny statements).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **deny ip host** *source* **host** *source*
4. **access-list** *access-list-number* **permit ip** *source*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **deny ip host** *source* **host** *source* **Example:** `Router(config)# access-list 101 deny ip host 10.0.0.1 host 10.0.0.2`	Defines a denied IP access list.
Step 4	**access-list** *access-list-number* **permit ip** *source* **Example:** `Router(config)# access-list 103 permit ip 209.165.200.225 0.255.255.255 10.20.0.0 0.255.255.255`	Defines an allowed IP access list.

What to Do Next

The access list defined in Step 4 is the same one that should be used to configure the SA. See the "Configuring the IPsec SA" section.

Configuring an IPsec Lifetime Timer

To configure an IPsec lifetime timer for a profile, perform the following steps. If this configuration task is not performed, the default is the maximum IPsec SA lifetime of 3600 seconds. The TEK lifetime value should be more than 900 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*

4. **set security-association lifetime seconds** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile profile1`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers and enters crypto ipsec profile configuration mode.
Step 4	**set security-association lifetime seconds** *seconds* **Example:** `Router(ipsec-profile)# set security-association lifetime seconds 2700`	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPsec SAs.

What to Do Next

Configure the IPsec SA. See the “Configuring IPsec SA” section.

Configuring an ISAKMP Lifetime Timer

To configure an ISAKMP lifetime timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **lifetime** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Defines an IKE policy and enters ISAKMP policy configuration mode.
Step 4	**lifetime** *seconds* **Example:** `Router(config-isakmp-policy)# lifetime 86400`	Specifies the lifetime of an IKE SA.

Configuring the IPsec SA

If time-based antireplay is configured on the key server but the group member is not capable of supporting it, the GDOI-3-GM_NO_CRYPTO_ENGINE syslog message is logged to the group member. See the "Cisco Group Encrypted Transport VPN System Logging Messages" section for a list of system error messages.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

To configure the IPsec SA, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* *transform* [*transform2...transform4*]
4. **crypto ipsec profile** *ipsec-profile-name*
5. **set transform-set** *transform-set-name*
6. **exit**
7. **crypto gdoi group** *group-name*
8. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
9. **server local**
10. **sa ipsec** *sequence-number*
11. **profile** *ipsec-profile-name*
12. **match address ipv4** {*access-list-number* | *access-list-name*}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*] **Example:** `Router(config)# crypto ipsec transform-set gdoi-trans esp-aes esp-sha-hmac`	Defines a transform set--an acceptable combination of security protocols and algorithms.
Step 4	**crypto ipsec profile** *ipsec-profile-name* **Example:** `Router(config)# crypto ipsec profile profile1`	Defines an IPsec profile and enters crypto ipsec profile configuration mode.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router(ipsec-profile)# set transform-set transformset1`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**exit** **Example:** `Router(ipsec-profile)# exit`	Exits IPsec profile configuration mode.
Step 7	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 8	Enter one of the following commands: • **identity number** *number* • **identity address ipv4** *address* **Example:** `Router(config-gdoi-group)# identity number 3333` **Example:**	Identifies a GDOI group number or address.

	Command or Action	Purpose
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`	
Step 9	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 10	**sa ipsec** *sequence-number* **Example:** `Router(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 11	**profile** *ipsec-profile-name* **Example:** `Router(gdoi-sa-ipsec)# profile gdoi-p`	Defines the IPsec SA policy for a GDOI group.
Step 12	**match address ipv4** {*access-list-number* | *access-list-name*} **Example:** `Router(gdoi-sa-ipsec)# match address ipv4 102`	Specifies an IP extended access list for a GDOI registration.
Step 13	**end** **Example:** `Router(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

What to Do Next

Replay should be configured. If replay is not configured, the default is counter mode.

Configuring Time-Based Antireplay for a GDOI Group

To configure time-based antireplay for a GDOI group, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *policy-name*
5. **server local**
6. **address** *ip-address*
7. **sa ipsec** *sequence-number*
8. **profile** *ipsec-profile-name*

9. **match address** {**ipv4** *access-list-number* | *access-list-name*}
10. **replay counter window-size** *seconds*
11. **replay time window-size** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroup1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *policy-name* **Example:** `Router(config-gdoi-group)# identity number 1234`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**address** *ip-address* **Example:** `Router(config-server-local)# address 209.165.200.225`	Sets the source address, which is used as the source for packets originated by the local key server.
Step 7	**sa ipsec** *sequence-number* **Example:** `Router(config-server-local)# sa ipsec 1`	Specifies the IPsec SA and enters GDOI SA IPsec configuration mode.
Step 8	**profile** *ipsec-profile-name* **Example:** `Router(gdoi-sa-ipsec)# profile test1`	Defines the IPsec SA policy for a GDOI group.

	Command or Action	Purpose
Step 9	**match address** {**ipv4** *access-list-number* | *access-list-name*} **Example:** `Router(gdoi-sa-ipsec)# match address ipv4 101`	Specifies an IP extended access list for a GDOI registration.
Step 10	**replay counter window-size** *seconds* **Example:** `Router(gdoi-sa-ipsec)# replay counter window-size 512`	Turns on counter-based antireplay protection for traffic defined inside an access list using GDOI if there are only two group members in a group. **Note** The behavior caused by this command and that caused by the **replay time window-size** command are mutually exclusive. You can configure either one without configuring the other.
Step 11	**replay time window-size** *seconds* **Example:** `Router(gdoi-sa-ipsec)# replay time window-size 1`	Sets the window size for antireplay protection using GDOI if there are more than two group members in a group. **Note** The behavior caused by this command and that caused by the **replay counter window-size** command are mutually exclusive. You can configure either one without configuring the other.

Configuring Passive SA

To configure passive SA (to put the group member in passive mode), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity** *name*
5. **passive**
6. **server address ipv4** {*address* | *hostname*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group group1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity** *name* **Example:** `Router(config-gdoi-group)# identity 2345`	Sets the identity to the crypto map.
Step 5	**passive** **Example:** `Router(config-gdoi-group)# passive`	Puts the group member into passive mode.
Step 6	**server address ipv4** {*address* | *hostname*} **Example:** `Router(config-gdoi-group)# server address ipv4 209.165.200.225`	Specifies the address of the server that a GDOI group is trying to reach.

Resetting the Role of the Key Server

To reset the cooperative role of the primary key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi ks coop role**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi ks coop role** **Example:** `Router# clear crypto gdoi ks coop role`	Resets the cooperative role of the key server.

Configuring a Group Member

To configure a group member, perform the following subtasks:

Configuring the Group Name ID Key Server IP Address and Group Member Registration

To configure the group name, ID, key server IP address, and group member registration, perform the following steps. You can configure up to eight key server addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Do one of the following:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server address ipv4** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Do one of the following:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

	Command or Action	Purpose
Step 5	**server address ipv4** *address* **Example:** `Router(config-gdoi-group)# server address ipv4 209.165.200.225`	Specifies the address of the server a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.

What to Do Next

Configure a crypto map. See the "Creating a Crypto Map Entry" section.

Creating a Crypto Map Entry

To create a crypto map entry and associate a GDOI group to it, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **gdoi**
4. **set group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **gdoi** **Example:** `Router(config)# crypto map mymap 10 gdoi`	Enters crypto map configuration mode and creates or modifies a crypto map entry.
Step 4	**set group** *group-name* **Example:** `Router(config-crypto-map)# set group group1`	Associates the GDOI group to the crypto map.

What to Do Next

Apply the crypto map to an interface to which the traffic has to be encrypted. See the "Applying the Crypto Map to an Interface to Which the Traffic Must Be Encrypted" section.

Applying the Crypto Map to an Interface to Which the Traffic Must Be Encrypted

To apply the crypto map to an interface to which the traffic must be encrypted, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **crypto map** *map-name* **redundancy** *standby-group-name* **stateful**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**crypto map** *map-name* **redundancy** *standby-group-name* **stateful** **Example:** `Router(config-if)# crypto map map1`	Applies the crypto map to the interface.

Activating Fail-Close Mode

Fail-close mode prevents unencrypted traffic from passing through a group member before that member is registered with a key server.

To configure a crypto map to work in fail-close mode, perform the following steps:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **crypto map** *map-name* **gdoi fail-close**
4. **match address** {*access-list-number* | *access-list-name*}
5. **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **gdoi fail-close** **Example:** `Router(config)# crypto map map1 gdoi fail-close`	Specifies that the crypto map is to work in fail-close mode and enters crypto map fail-close configuration mode.
Step 4	**match address** {*access-list-number* | *access-list-name*} **Example:** `Router(crypto-map-fail-close)# match address 133`	(Optional) Specifies an ACL for a GDOI registration.
Step 5	**activate** **Example:** `Router(crypto-map-fail-close)# activate`	Activates fail-close mode.

Configure Fail Close Revert

Note Activating fail close mode is mandatory for the Fail Close Revert feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*

5. **server address ipv4** *address*
6. **client fail-close revert**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server address ipv4** *address* **Example:** `Router(config-gdoi-group)# server address ipv4 10.0.5.2`	Specifies the address of the server a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.
Step 6	**client fail-close revert** **Example:** `Router(config-gdoi-group)# client fail-close revert`	Enables the client fail close revert feature
Step 7	**end** **Example:** `Router(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring Acceptable Ciphers or Hash Algorithms for KEK

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

To configure the ciphers and hash algorithms for KEK to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server address ipv4** *address*
6. **client rekey encryption** *cipher* [... [*cipher*]]
7. **client rekey hash** *hash*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	Enter one of the following commands: • **identity number** *number* • **identity address ipv4** *address* **Example:** `Router(config-gdoi-group)# identity number 3333`	Identifies a GDOI group number or address.

	Command or Action	Purpose
	Example: `Router(config-gdoi-group)# identity address ipv4 10.2.2.2`	
Step 5	**server address ipv4** *address* **Example:** `Router(config-gdoi-group)# server address ipv4 10.0.5.2`	Specifies the address of the server a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.
Step 6	**client rekey encryption** *cipher* [... [*cipher*]] **Example:** `Router(config-gdoi-group)# client rekey encryption aes 128 aes 192 aes 256`	Sets the client acceptable rekey ciphers for the KEK.
Step 7	**client rekey hash** *hash* **Example:** `Router(config-gdoi-group)# client rekey hash sha`	Sets the client acceptable hash algorithm for KEK.
Step 8	**end** **Example:** `Router(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring Acceptable Transform Sets for TEK

To configure the transform sets used by TEKs for data encryption or authentication to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*]
4. **exit**
5. **crypto gdoi group** *group-name*
6. **client transform-sets** *transform-set-name1* [... [*transform-set-name6*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*] **Example:** `Router(config)# crypto ipsec transform-set g1 esp-aes 192 esp-sha-hmac`	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode.
Step 4	**exit** **Example:** `Router(cfg-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 5	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 6	**client transform-sets** *transform-set-name1* [... [*transform-set-name6*]] **Example:** `Router(config-gdoi-group)# client transform-sets g1`	Specifies the acceptable transform-set tags used by TEK for data encryption and authentication. • You can specify up to six transform-set tags.
Step 7	**end** **Example:** `Router(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Tracking the Group Member Crypto State

Perform this task to track the crypto state of the group member (GM) using the configured Enhanced Object Tracker (EOT) stub-object ID.

Before you begin

You must configure an Enhanced Object Tracking (EOT) by creating a stub-object and assign the object with a tracking ID to monitor the GDOI MIB. The following is a sample configuration in which, tracking ID 99 is assigned to the stub-object.

```
event manager applet test1
 event snmp oid <new GDOI MIB object> .....
```

```
 action 2.0 track set 99 state up

track 99 stub-object
 delay up 60
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **client status active-sa track** *tracking-number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**client status active-sa track** *tracking-number* **Example:** `Device(config-gdoi-group)# client status active-sa track 99`	Enables the tracking for the stub-object. In this example, a GM will set the stub-object 99 to state “UP” when it receives valid traffic encryption key (TEK) from the key server (KS).On the other hand, the GM will set the stub-object 99 to state “DOWN” if it has no valid TEK because of errors, such as registration failure or TEK expiration before rekey.
Step 5	**exit** **Example:** `Device(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring GET VPN GM Authorization

GET VPN GM authorization can be done using preshared keys or PKI. It is a best practice to turn on GET VPN authorization. When a key server serves multiple GDOI groups, key server authorization is required to prevent a GM in one group from requesting keys and policies from another group. The ISAKMP authorization confirms that the GM is allowed to request GDOI attributes from the key server while the GDOI authorization confirms the GM is allowed to request GDOI attributes from a specific group configured in the key server.

To configure GET VPN GM authorization, perform either of the following tasks:

Configuring GM Authorization Using Preshared Keys

To configure GM authorization using preshared keys, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **server local**
5. **authorization address ipv4 {** *access-list-name* | *access-list-number* **}**
6. **exit**
7. **exit**
8. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group getvpn`	Identifies a GDOI and enters GDOI group configuration mode.
Step 4	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 5	**authorization address ipv4 {** *access-list-name* | *access-list-number* **}** **Example:** `Router(gdoi-local-server)# authorization address ipv4 50`	Specifies a list of addresses for a GDOI.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI local configuration mode and returns to GDOI group configuration mode.
Step 7	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 8	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]] **Example:** `Router(config)# access-list 50 permit ip 209.165.200.225 0.0.0.0 209.165.200.254 0.0.0.0`	Defines an allowed IP access list. • In the example, an access list with access list number 50 is defined, and packets sent from source IP address 209.165.200.225 to destination IP address 209.165.200.254 are permitted.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring GM Authorization Using PKI

To configure GM authorization using PKI, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp identity** {**address** | **dn** | **hostname**}
4. **crypto pki trustpoint** *name*
5. **subject-name** [*x.500-name*]
6. **exit**
7. **crypto gdoi group** *group-name*
8. **server local**
9. **authorization identity** *name*
10. **exit**
11. **exit**
12. **crypto identity** *name*
13. **dn** *name=string* [, *name=string*]
14. **exit**

15. **crypto isakmp identity** {**address** | **dn** | **hostname** }
16. **crypto pki trustpoint** *name*
17. **subject-name** [*x.500-name*]
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp identity** {**address**	**dn**	**hostname**} **Example:** `Router(config)# crypto isakmp identity dn`	Defines the identity used by the router when the router is participating in the Internet Key Exchange (IKE) protocol.
Step 4	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint GETVPN`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name OU=GETVPN`	Specifies the subject name in the certificate request.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group getvpn`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 8	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.

	Command or Action	Purpose
Step 9	**authorization identity** *name* **Example:** `Router(gdoi-local-server)# authorization identity GETVPN_FILTER`	Specifies an identity for a GDOI group.
Step 10	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to GDOI group configuration mode.
Step 11	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 12	**crypto identity** *name* **Example:** `Router(config)# crypto identity GETVPN_FILTER`	Configures the identity of the router with a given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 13	**dn** *name=string* [, *name=string*] **Example:** `Router(config-crypto-identity)# dn ou=GETVPN`	Associates the identity of a router with the DN in the certificate of the router.
Step 14	**exit** **Example:** `Router(config-crypto-identity)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 15	**crypto isakmp identity** {**address** | **dn** | **hostname** } **Example:** `Router(config)# crypto isakmp identity dn`	Defines the identity used by the router when the router is participating in the IKE protocol.
Step 16	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint GETVPN`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 17	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name ou=getvpn`	Specifies the subject name in the certificate request.

	Command or Action	Purpose
Step 18	**end** **Example:** `Router(ca-trustpoint)# exit`	Exits GDOI group configuration mode, saves the configuration, and returns to privileged EXEC mode.

Verifying and Troubleshooting Cisco Group Encrypted Transport VPN Configurations

The following tasks can be used to verify and troubleshoot your GET VPN configurations. These tasks are optional and are used to gather information during troubleshooting.

Note With CSCsi82594, if Time-based Anti-Replay (TBAR) is enabled, the rekey time period is set to 2 hours (7200 seconds). In this scenario, the Key Server periodically sends a rekey to the Group Members every 2 hours (7200 seconds). In the below example, even though the Traffic Encryption Key (TEK) lifetime is set to 28800 seconds (8 hours), the rekey timer is still 2 hours. For show outputs displaying TBAR information, use the **show crypto gdoi gm replay** and **show crypto gdoi ks replay** commands.

```
crypto ipsec profile atm-profile
set security-association lifetime seconds 28800
!
crypto gdoi group ATM-DSL
server local
   sa ipsec 1
      !
      replay time window-size 100
```

Verifying Active Group Members on a Key Server

To verify active group members on a key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi ks members**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show crypto gdoi ks members** **Example:** `Router# show crypto gdoi ks members`	Displays information about key server members.

Verifying Rekey-Related Statistics

To verify rekey-related statistics, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi ks rekey**
3. **show crypto gdoi [gm]**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show crypto gdoi ks rekey**

Example:

```
Device# show crypto gdoi ks rekey

For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):

# of teks : 1 Seq num : 0
KEK POLICY (transport type : Unicast)
spi : 0xA8110DE7CC8B0FB201F2A8BFAA0F2D90
management alg : disabled encrypt alg : 3DES
crypto iv length : 8 key size : 24
orig life(sec): 300 remaining life(sec): 296 <------- ticking down
sig hash algorithm : enabled sig key length : 94
sig size : 64
sig key name : mykeys
```

On the key server, this command displays information about the rekeys that are being sent from the key server. The output displays the ticking down of the KEK remaining lifetime.

Step 3 **show crypto gdoi [gm]**

Example:

```
Device# show crypto gdoi
```

```
GROUP INFORMATION

Group Name : diffint
Group Identity : 3333
Rekeys received : 0
IPSec SA Direction : Both

Group Server list : 10.0.8.1

Group member : 10.0.3.1 vrf: None
Version : 1.0.2
Registration status : Registered
Registered with : 10.0.8.1
Re-registers in : 93 sec <-----------re-registration time for TEK or KEK
Succeeded registration: 1
Attempted registration: 1
Last rekey from : 0.0.0.0
Last rekey seq num : 0
Unicast rekey received: 0
Rekey ACKs sent : 0
Rekey Received : never
allowable rekey cipher: any
allowable rekey hash : any
allowable transformtag: any ESP

Rekeys cumulative
Total received : 0
After latest register : 0
Rekey Acks sents : 0

ACL Downloaded From KS 10.0.8.1:
access-list permit ip host 10.0.1.1 host 239.0.1.1
access-list permit ip host 10.0.100.2 host 238.0.1.1

KEK POLICY:
Rekey Transport Type : Unicast
Lifetime (secs) : 255 <-----------lifetime ticking
Encrypt Algorithm : 3DES
Key Size : 192
Sig Hash Algorithm : HMAC_AUTH_SHA
Sig Key Length (bits) : 512
```

On the group member, this command displays information about the rekeys that are being sent from the key server. The "re-registers in" field of the output displays the duration after which the group member re-registers for a TEK or a KEK, whichever time is smaller

Verifying IPsec SAs That Were Created by GDOI on a Group Member

To verify IPsec SAs that were created by GDOI on a group member, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi group** *group-name* **ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi group** *group-name* **ipsec sa** **Example:** `Router# show crypto gdoi group diffint ipsec sa`	Displays information about IPsec SAs that were created by GDOI on a group member. • In this case, information will be displayed only for group "diffint." • For information about IPsec SAs for all groups, omit the **group** keyword and *group-name* argument.

Verifying IPsec SAs That Were Created by GDOI on a Key Server

To verify IPsec SAs that were created by GDOI on a key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec sa** **Example:** `Device# show crypto ipsec sa`	Displays the settings used by current SAs.

Verifying the TEKs that a Group Member Last Received from the Key Server

To verify the TEKs that a GM last received from the KS, perform the following steps on the GM:

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi** **Example:** `Router# show crypto gdoi`	Displays the current GDOI configuration and the policy that is downloaded from the KS. The TEKs are listed in the TEK POLICY section. Without enabling debugging, you can use this command to compare the TEKs that a GM actually last received with the TEKs downloaded from the KS to the IPsec control plane (which you can view using the **show crypto ipsec sa** command).

Verifying Cooperative Key Server States and Statistics

To verify cooperative key server states and statistics, perform the following steps, using one or both of the **debug** and **show** commands shown.

SUMMARY STEPS

1. **enable**
2. **debug crypto gdoi ks coop**
3. **show crypto gdoi group** *group-name* **ks coop** [**version**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto gdoi ks coop** **Example:** `Router# debug crypto gdoi ks coop`	Displays information about a cooperative key server.
Step 3	**show crypto gdoi group** *group-name* **ks coop** [**version**] **Example:** `Router# show crypto gdoi group diffint ks coop version`	Displays information for the group “diffint” and version information about the cooperative key server.

Verifying Antireplay Pseudotime-Related Statistics

To verify antireplay pseudotime-related statistics, perform the following steps using one or all of the **clear**, **debug**, and **show** commands.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi group** *group-name* **replay**
3. **debug crypto gdoi replay**
4. **show crypto gdoi group** *group-name*
5. **show crypto gdoi group** *group-name* **ks replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi group** *group-name* **replay** **Example:** `Router# clear crypto gdoi group diffint replay`	Clears the replay counters.
Step 3	**debug crypto gdoi replay** **Example:** `Router# debug crypto gdoi replay`	Displays information about the pseudotime stamp that is contained in a packet.
Step 4	**show crypto gdoi group** *group-name* **Example:** `Router# show crypto gdoi group diffint`	Displays information about the current pseudotime of the group member. • It also displays the different counts that are related to the antireplay for this group.
Step 5	**show crypto gdoi group** *group-name* **ks replay** **Example:** `Router# show crypto gdoi group diffint ks replay`	Displays information about the current pseudotime of the key server.

Verifying the Fail-Close Mode Status of a Crypto Map

To verify the fail-close mode status of a crypto map, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **show crypto map gdoi fail-close**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto map gdoi fail-close** **Example:** `Router# show crypto map gdoi fail-close`	Displays information about the status of the fail-close mode.

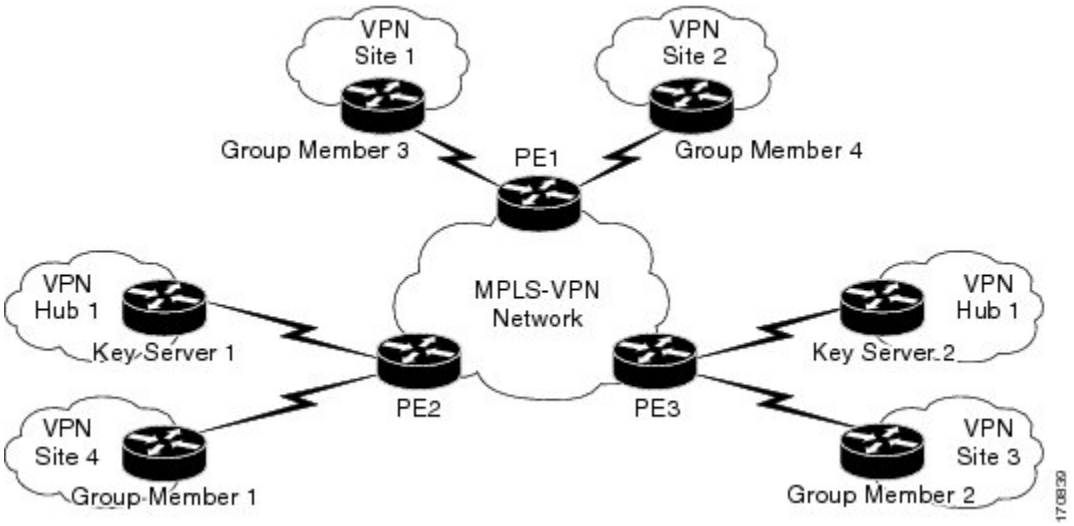
Configuration Examples for Cisco Group Encrypted Transport VPN

Example: Key Server and Group Member Case Study

The following case study includes encrypting traffic CE-CE in an MPLS VPN environment.

The MPLS VPN core interconnects VPN sites as is shown in the figure below. VPN site CPEs, Group Member 1 through Group Member 4, are grouped into a single GDOI group that correlates with a VPN with which these sites are a part. This scenario is an intranet VPN scenario. All the key servers and Group Members are part of the same VPN. Key Server 1 and Key Server 2 are the cooperative key servers that support VPN members Group Member 1 through Group Member 4. Key Server 1 is the primary key server and Key Server 2 is the secondary key server.

Figure 125: Key Server and Group Member Scenario



The following configuration examples are based on the case study in the figure above.

Example Key Server 1

Key server 1 is the primary key server.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname KS1
!
logging buffered 100000 debugging
no logging console
!
no aaa new-model
!
resource policy
!
clock timezone EST 0
ip subnet-zero
no ip domain lookup
ip domain name cisco.com
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 lifetime 400
crypto isakmp key cisco address 10.1.1.13
crypto isakmp key cisco address 10.1.1.9
crypto isakmp key cisco address 10.1.1.1
crypto isakmp key cisco address 10.1.1.5
crypto isakmp key cisco address 10.1.1.21
!
crypto ipsec transform-set gdoi-trans-group1 esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-profile-group1
 set security-association lifetime seconds 1800
 set transform-set gdoi-trans-group1
!
crypto gdoi group group1
 identity number 1
 server local
  rekey lifetime seconds 86400
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa group1-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-group1
   match address ipv4 101
   replay counter window-size 64
  address ipv4 209.165.200.225
  redundancy
   local priority 10
   peer address ipv4 209.165.200.225
   !
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
!
ip classless
```

```
ip route 0.0.0.0 0.0.0.0 10.1.1.18
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
end
```

Example Key Server 2

Key Server 2 is the secondary key server.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname KS2
!
logging buffered 100000 debugging
no logging console
!
no aaa new-model
!
resource policy
!
clock timezone EST 0
ip subnet-zero
no ip domain lookup
ip domain name cisco
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 lifetime 400
crypto isakmp key cisco address 10.1.1.9
crypto isakmp key cisco address 10.1.1.1
crypto isakmp key cisco address 10.1.1.5
crypto isakmp key cisco address 10.1.1.17
crypto isakmp key cisco address 10.1.1.13
!
crypto ipsec transform-set gdoi-trans-group1 esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-profile-group1
 set security-association lifetime seconds 1800
 set transform-set gdoi-trans-group1
!
crypto gdoi group group1
 identity number 1
 server local

  rekey lifetime seconds 86400
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa group1-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-group1
   match address ipv4 101
   replay counter window-size 64
  address ipv4 10.1.1.21
  redundancy
   local priority 1
   peer address ipv4 10.1.1.17
```

```
   !
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.22
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
end
```

Example: Configuring Group Member 1

Group member 1 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GM1
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
 crypto map map-group1
!
router bgp 1000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 neighbor 10.1.1.2 remote-as 5000
 no auto-summary
!
ip classless
!
End
```

The same GDOI group cannot be applied to multiple interfaces. The following examples show unsupported cases:

Example 1

```
crypto map map-group1
  group g1
 interface ethernet 1/0
  crypto map map-group1
 interface ethernet 2/0
  crypto map map-group1
```

Example 2

```
crypto map map-group1 10 gdoi
  set group group1
crypto map map-group2 10 gdoi
  set group group1
 interface ethernet 1/0
  crypto map map-group1
 interface ethernet 2/0
```

Example: Configuring Group Member 2

Group member 2 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname GM2
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.201.1
 server address ipv4 209.165.200.225
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
 crypto map map-group1
!
router bgp 2000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.2.0 mask 255.255.255.0
 neighbor 10.1.1.6 remote-as 5000
 no auto-summary
!
```

```
ip classless
!
end
```

Example: Configuring Group Member 3

Group member 3 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GM3
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto ipsec transform-set gdoi-trans-group1 esp-aes esp-sha-hmac
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.252
 crypto map map-group1
!
router bgp 3000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.3.0 mask 255.255.255.0
 neighbor 10.1.1.10 remote-as 5000
 no auto-summary
!
ip classless
!
end
```

Example: Configuring Group Member 4

Group member 4 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname GM4
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.252
 crypto map map-group1
!
router bgp 4000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.4.0 mask 255.255.255.0
 neighbor 10.1.1.14 remote-as 5000
 no auto-summary
!
ip classless
!
end
```

Example: Configuring Group Member 5

If a group member has multiple interfaces that are part of the same GDOI group, you should use a loopback interface to source the crypto. If a loopback interface is not used, each interface that handles encrypted traffic must register individually with the key server.

The key server sees these as separate requests and must keep multiple records for the same group member, which also means sending multiple rekeys. If crypto is sourced from the loopback interface instead, the group member registers only once with the key server.

The following configuration shows how the group member registers once with the key server:

```
!


interface GigabitEthernet0/1
 description *** To AGG-1 ***
 crypto map dgvpn
!
interface GigabitEthernet0/2
 description *** To AGG-2 ***
 crypto map dgvpn
!
interface Loopback0
 ip address 209.165.201.1 255.255.255.255
!
```

```
 crypto map dgvpn local-address Loopback0
!
```

Example: Verifying the TEKs That a Group Member Last Received from the Key Server

The following example shows how to display the current GDOI configuration and the policy that is downloaded from the KS:

```
Device# show crypto gdoi

GROUP INFORMATION

    Group Name               : GETV6
.
.
.
KEK POLICY:
.
.
.
TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet2/0:
    IPsec SA:
        spi: 0x627E4B84(1652444036)
        transform: esp-aes
        sa timing:remaining key lifetime (sec): (3214)
        Anti-Replay(Time Based) : 10 sec interval
        tag method : cts sgt
        alg key size: 24 (bytes)
        sig key size: 20 (bytes)
        encaps: ENCAPS_TUNNEL


GROUP INFORMATION

    Group Name               : GETV4
.
.
.
KEK POLICY:
.
.
.
TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet2/0:
    IPsec SA:
        spi: 0xF6E6B597(4142314903)
        transform: esp-aes
        sa timing:remaining key lifetime (sec): (3214)
        Anti-Replay : Disabled
        tag method : cts sgt
        alg key size: 24 (bytes)
        sig key size: 20 (bytes)
        encaps: ENCAPS_TUNNEL
```

The TEKs are listed in the TEK POLICY section. Without enabling debugging, you can use this command to compare the TEKs that a GM actually last received with the TEKs downloaded from the KS to the IPsec control plane (which you can view using the **show crypto ipsec sa** command).

The tag method field shows the method used for GET VPN inline tagging; the possible values are either cts sgt (for Cisco TrustSec security group tags) or disabled. The alg key size field shows the key length for the encryption algorithm that is configured in the TEK policy. The sig key size field shows the key length for the signature that is configured in the TEK policy. The encaps field shows the type of IPsec encapsulation (either tunnel or transport) that is configured in the TEK policy.

The output from this command might show that a TEK has expired since the time it was received from the KS.

Example Passive SA

The following example displays information about crypto rules on outgoing packets:

```
Router# show crypto ruleset
Ethernet0/0:
  59 ANY ANY DENY
  11 ANY/848 ANY/848 DENY
 IP ANY ANY IPSec SA Passive
 IP ANY ANY IPSec Cryptomap
```

The following example displays the directional mode of the IPsec SA:

```
Router# show crypto ruleset detail
Ethernet0/0:
 20000001000019  59 ANY ANY DENY -> 20000001999999
20000001000029  11 ANY/848 ANY/848 DENY -> 20000001999999
 20000001000035 IP ANY ANY IPSec SA Passive
 20000001000039 IP ANY ANY IPSec Cryptomap
```

Example Fail-Close Mode

The following example shows that fail-close mode has been activated, and unencrypted traffic from access list 102 is allowed before the group member is registered:

```
crypto map map1 gdoi fail-close
 match address 102
 activate
crypto map map1 10 gdoi
 set group ks1_group
 match address 101
!
access-list 101 deny ip 10.0.1.0 0.0.0.255 10.0.1.0 0.0.0.255
access-list 102 deny tcp any eq telnet any
```

The following **show crypto map gdoi fail-close** command output shows that fail-close has been activated:

```
Router# show crypto map gdoi fail-close

Crypto Map: "svn"
```

```
Activate: yes
Fail-Close Access-List: (Deny = Forward In Clear, Permit = Drop)
    access-list 105 deny tcp any port = 23 any
    access-list 105 deny ospf any any
```

Example: Verifying Fail-Close Revert

```
Device#show cry gdoi group GDOI_GROUP_1 | i Fail|Policy
       Fail-Close Revert : Enabled
       KS Policy Removal in : 697 sec
```

Additional References for Cisco Group Encrypted Transport VPN

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
CISCO-GDOI-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco Group Encrypted Transport VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 289: Feature Information for Cisco Group Encrypted Transport VPN

Feature Name	Releases	Feature Information
Cisco Group Encrypted Transport VPN	Cisco IOS XE Release 2.3	Cisco Group Encrypted Transport VPN is an optimal encryption solution for large-scale IP or MPLS sites that require any-to-any connectivity with minimum convergence time, low processing, provisioning, managing, and troubleshooting overhead. The following commands were introduced or modified: **address ipv4 (GDOI)**, **clear crypto gdoi**, **crypto gdoi gm**, **debug crypto gdoi**, **local priority**, **peer address ipv4**, **redundancy**, **rekey address ipv4**, **rekey transport unicast**, **replay counter window-size**, **replay time window-size**, **sa receive-only**, **show crypto gdoi**.
Create MIB Object to Track a Successful GDOI Registration	Cisco IOS XE Release 3.12S	The Create MIB Object to track a successful GDOI Registration feature introduces a new MIB object in the GDOI MIB to indicate the number of active TEKs in a group.

Feature Name	Releases	Feature Information
GET VPN Hardening	Cisco IOS XE Release 3.9S	This feature improves GET VPN resiliency. The improvements in resiliency prevent or minimize data-traffic disruption by using one of the following methods: • Making corrections when conditions that could cause a traffic disruption are detected. • Rapidly executing a recovery mechanism when a disruption is detected. The following commands were modified: **show crypto gdoi**, **show crypto ipsec sa**, **show tech-support**.
GET VPN IKEv1 Separation	Cisco IOS XE Release 3.11S	This feature eases maintenance and troubleshooting. The following commands were modified: **show tech-support**, **show crypto gdoi** and **show crypto ipsec sa**.
GET VPN Phase 1.2	Cisco IOS XE Release 2.3	These enhancements include the following features: • Change Key Server Role This feature enables you to change the role of the key server from primary to secondary. The following commands were added or modified for this feature: **clear crypto gdoi ks coop role** • Fail-Close Mode This feature prevents unencrypted traffic from passing through the group member before that member is registered. The following commands were added or modified for this feature: **activate**, **crypto map**, **match address**, and **show crypto map**. • Passive SA This feature allows a group member to be configured into passive mode permanently. The following command was introduced: **passive**.
GETVPN Routing Awareness for BGP	Cisco IOS XE Release 3.13S	The following commands were introduced or modified: **client status active-sa track**.

Feature Name	Releases	Feature Information
GET VPN Resiliency	Cisco IOS XE Release 3.9S	This feature improves the resiliency of GET VPN, so that data traffic disruption is prevented or minimized when errors occur. This feature introduces long SA lifetime functionality, which extends the maximum for which you can configure the lifetime of the key encryption key and traffic encryption keys from 24 hours to 30 days. This feature also lets you configure key servers to continue to send periodic reminder rekeys to group members that did not respond with an acknowledgment in the last scheduled rekey. By using a long SA lifetime in combination with periodic reminder rekeys, a key server can effectively synchronize group members if they miss a scheduled rekey before the keys roll over. The following commands were modified: **rekey lifetime**, **rekey retransmit**, **set security-association lifetime**, **show crypto gdoi**.
GET VPN Support of IPsec Inline Tagging for Cisco TrustSec	Cisco IOS XE Release 3.9S	Cisco TrustSec (CTS) uses the user and device identification information acquired during authentication to classify packets as they enter the network. CTS maintains classification of each packet by tagging packets with security group tags (SGTs) on ingress to the CTS network so that they can be identified for applying security and other policy criteria along the data path. The tags allow network intermediaries such as switches and firewalls to enforce the access control policy based on the classification. The GET VPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GET VPN inline tagging to carry the SGT information across the private WAN. The following commands were introduced or modified: **show crypto gdoi**, **show crypto ipsec sa**, **tag cts sgt**.
GET VPN Time-Based Anti-Replay	Cisco IOS XE Release 2.3	Support for time-based antireplay was added to the Cisco VSA.
GET VPN Troubleshooting	Cisco IOS XE Release 3.8S	This feature provides improved debugging levels (so debug messages can be enabled per feature), event logging, exit trace capabilities to save a log of error conditions and their tracebacks, and conditional debugging (which provides the ability to debug individual group members from the key server). The conditional debugging feature provides the ability to perform conditional debugging on the key server so that it can filter based on GM or other cooperative key servers. The event logging feature provides the ability to log the last set of events. The following commands were introduced or modified: **clear crypto gdoi**, **debug crypto condition unmatched**, **debug crypto gdoi**, **debug crypto gdoi condition**, **monitor event-trace gdoi**, **show crypto gdoi**, and **show monitor event-trace gdoi**.

Feature Name	Releases	Feature Information
Group Encrypted Transport VPN Key Server	Cisco IOS XE Release 3.6S	Support was added for configuring a device running Cisco IOS XE as a key server. In Cisco IOS XE Release 3.6S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.13S, support was added for the Cisco Cloud Services Router (CSR) 1000V Series.
VSA Support for GET VPN	Cisco IOS XE Release 2.3	Cisco VSA (high-performance crypto engine) support was added for GDOI and GET VPN.

Glossary

DOI—Domain of Interpretation. For Internet Security Association Key Management Protocol (ISAKMP), a value in the security association (SA) payload that describes in which context the key management message is being sent (IPsec or Group Domain of Interpretation).

GDOI—Group Domain of Interpretation. For ISAKMP, a means of distributing and managing keys for groups of mutually trusted systems.

group member—Device (Cisco IOS router) that registers with a group that is controlled by the key server for purposes of communicating with other group members.

group security association—SA that is shared by all group members in a group.

IPsec—IP security. Data encryption protocol for IP packets that are defined in a set of RFCs (see IETF RFC 2401).

ISAKMP—Internet Security Association and Key Management Protocol. Protocol that provides a framework for cryptographic key management protocols.

KEK—key encryption key. Key used to protect the rekey between the key server and group members.

key server—Device (Cisco IOS router) that distributes keys and policies to group members.

MTU—maximum transmission unit. Size (in bytes) of the largest packet or frame that a given layer of a communications protocol can pass onward.

SA—security association. SA that is shared by all group members in a group.

Simple Network Management Protocol (SNMP)—An interoperable standards-based protocol that allows for external monitoring of a managed device through an SNMP agent.

TEK—traffic encryption key. Key that is used to protect the rekey between group members.

CHAPTER 220

GET VPN GM Removal and Policy Trigger

The GET VPN GM Removal and Policy Trigger feature lets you easily remove unwanted group members (GMs) from the group encrypted transport (GET) VPN network, provides a rekey triggering method to install new security associations (SAs) and remove obsolete SAs, and lets you check whether devices are running versions of GET VPN software that support these capabilities.

Information About GM Removal and Policy Trigger

GET VPN Software Versioning

GET VPN software versions are of the form

major-version.minor-version.mini-version

where

- *major-version* defines compatibility for all GET VPN devices.
- *minor-version* defines compatibility for key server (KS)-to-KS (cooperative key server) associations and for GM-to-GM interoperability.
- *mini-version* tracks feature changes that have no compatibility impact.

For example, the base version (for all prior GET VPN features) is 1.0.1. Also, for example, the version that contains the GM removal feature and the policy replacement feature is 1.0.2, which means that these features are fully backward compatible with the base version (despite the introduction of behavior in these features for triggered rekeys).

GMs send the GET VPN software version to the KS in the vendor-ID payload during Internet Key Exchange (IKE) phase 1 negotiation (which is defined in RFC 2408, *Internet Security Association and Key Management Protocol [ISAKMP]*). KSs send the software version to other cooperative KSs in the version field of the cooperative KS announcement (ANN) messages. Cooperative KSs also synchronize their lists of versions that each GM is using.

The GM removal feature and the policy replacement feature each provide a command that you run on the KS (or primary KS) to find devices in the group that do not support that feature.

GM Removal

Without the GM removal and policy replacement features, you would need to complete the following steps to remove unwanted GMs from a group:

1. Revoke the phase 1 credential (for example, the preshared key or one or more PKI certificates).
2. Clear the traffic encryption key (TEK) and key encryption key (KEK) database on the KS.
3. Clear the TEK and KEK database on each GM individually and force each GM to re-register.

The third step is time-consuming when a GET VPN group serves thousands of GMs. Also, clearing the entire group in a production network might cause a network disruption. The GET VPN GM Removal and Policy Trigger feature automates this process by introducing a command that you enter on the KS (or primary KS) to create a new set of TEK and KEK keys and propagate them to the GMs.

GM Removal Compatibility with Other GET VPN Software Versions

You should use the GET VPN GM Removal and Policy Trigger feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. Otherwise, secondary KSs or GMs running older software will ignore the GM removal message and continue to encrypt and decrypt traffic using the old SAs. This behavior causes network traffic disruption.

This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support GM removal. When the primary KS tries to remove GMs in a network containing devices that do not support GM removal, a warning message appears. For more information, see the “Ensuring That GMs Are Running Software Versions That Support GM Removal” section.

GM Removal with Transient IPsec SAs

The GET VPN GM Removal and Policy Trigger feature provides a command that you use on the KS (or primary KS) to trigger GM removal with transient IPsec SAs. This behavior shortens key lifetimes for all GMs and causes them to re-register before keys expire. During GM removal, no network disruption is expected, because traffic continues to be encrypted and decrypted using the transient IPsec SA until its lifetime expires. For more information, see the “Removing GMs with Transient IPsec SAs” section.

GM Removal with Immediate IPsec SA Deletion

The GET VPN GM Removal and Policy Trigger feature provides an optional keyword that you can use on the KS (or primary KS) to force GMs to delete old TEKs and KEKs immediately (without using transient SAs) and re-register. However, this behavior can cause a disruption to the data plane, so you should use this method only for important security reasons. For more information, see the “Removing GMs and Deleting IPsec SAs Immediately” section.

Policy Replacement and Rekey Triggering

The GET VPN GM Removal and Policy Trigger feature provides a new rekey triggering method to remove obsolete SAs and install new SAs.

Inconsistencies Regarding Which TEK and KEK Policy Changes Will Trigger Rekeys

Without this feature, there are inconsistencies regarding which TEK and KEK policy changes will trigger rekeys:

- Multiple rekeys could be sent during the course of security policy updates.
- Some policy changes (for example, transform set, profile, lifetime, and anti-replay) will install new SAs on GMs; however, the SAs from the existing policies remain active until their lifetimes expire.
- Some policy changes (for example, a TEK's access control entry/access control list (ACE/ACL) changes) will install new SAs on GMs and take effect immediately. However, the obsolete SAs are kept in each GM's database (and can be displayed using the **show crypto ipsec sa** command until their lifetimes expire).

For example, if the KS changes the policy from Data Encryption Standard (DES) to Advanced Encryption Standard (AES), when the GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). The GM continues to encrypt and decrypt traffic using the old SAs until their shortened lifetimes expire.

Following is the formula to calculate the shortened lifetime:

TEK_SLT = MIN(TEK_RLT, MAX(90s, MIN(5%(TEK_CLT), 3600s)))

where

- TEK_SLT is the TEK shortened lifetime
- TEK_RLT is the TEK remaining lifetime
- TEK_CLT is the TEK configured lifetime

The following table summarizes the inconsistencies regarding rekeys.

Table 290: Rekey Behavior After Security Policy Changes

Policy Changes	Rekey Sent?	Rekey Behavior After Policy Changes
TEK: SA lifetime	No	The old SA remains active until its lifetime expires. The new lifetime will be effective after the next scheduled rekey. Even if you enter the **clear crypto sa** command, it will re-register and download the old SA with the old lifetime again.
TEK: IPSEC transform set	Yes	The SA of the old transform set remains active until its lifetime expires.
TEK: IPSEC profile	Yes	The SA of the old profile remains active until its lifetime expires.
TEK: Matching ACL	Yes	Outbound packet classification immediately uses the ACL. But the old SAs remain in the SA database (you can view them by using the **show crypto ipsec sa** command).
TEK: Enable replay counter	Yes	But the old SA without counter replay remains active until its lifetime expires.

Policy Changes	Rekey Sent?	Rekey Behavior After Policy Changes
TEK: Change replay counter value	No	The SA with a new replay counter is sent out in the next scheduled rekey.
TEK: Disable replay counter	Yes	But the old SA with counter replay enabled remains active until its lifetime expires.
TEK: Enable TBAR	Yes	But the old SA with time-based anti-replay (TBAR) disabled remains active until its lifetime expires.
TEK: Change TBAR window	No	The SA with a new TBAR window will be sent out in the next scheduled rekey.
TEK: Disable TBAR	Yes	But the old SA with TBAR enabled remains active until its lifetime expires.
TEK: Enable receive-only	Yes	Receive-only mode is activated right after the rekey.
TEK: Disable receive-only	Yes	Receive-only mode is deactivated right after the rekey.
KEK: SA lifetime behavior	No	The change is applied with the next rekey.
KEK: Change authentication key	Yes	The change is applied immediately.
KEK: Change crypto algorithm	Yes	The change is applied immediately.

This feature solves these problems by ensuring consistency. With this feature, GET VPN policy changes alone will no longer trigger a rekey. When you change the policy (and exit from global configuration mode), a syslog message appears on the primary KS indicating that the policy has changed and a rekey is needed. This feature provides a new command that you then enter on the KS (or primary KS) to send a rekey (that is based on the latest security policy in the running configuration).

This feature also provides an extra keyword to the new command to force a GM receiving the rekey to remove the old TEKs and KEK immediately and install the new TEKs and KEK. Therefore, the new policy takes effect immediately without waiting for old policy SAs to expire. (However, using this keyword could cause a temporary traffic discontinuity, because all GMs might not receive the rekey message at the same time.)

Policy Replacement and Rekey Triggering Compatibility with Other GET VPN Software Versions

You should use rekey triggering only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. For GMs running older versions that do not yet support the **crypto gdoi ks** command, the primary KS uses the software versioning feature to detect those versions and only triggers a rekey without sending instruction for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs. (This behavior is the same as the prior rekey method and ensures backward compatibility for devices that cannot support policy replacement.)

This feature provides a command that you use on the KS (or primary KS) to check whether all the devices in the network are running versions that support policy replacement. For more information, see the “Ensuring That GMs Are Running Software Versions That Support Policy Replacement” section.

How to Configure GET VPN GM Removal and Policy Trigger

Ensuring That GMs Are Running Software Versions That Support GM Removal

You should use the GET VPN GM Removal and Policy Trigger feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. Otherwise, secondary KSs or GMs that are running older software will ignore the GM removal message and continue to encrypt and decrypt traffic using the old SAs. This behavior causes network traffic disruption.

Perform this task on the KS (or primary KS) to ensure that all devices in the network support GM removal.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature gm-removal**
3. **show crypto gdoi feature gm-removal | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature gm-removal** **Example:** `Device# show crypto gdoi feature gm-removal`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports GM removal.
Step 3	**show crypto gdoi feature gm-removal | include No** **Example:** `Device# show crypto gdoi feature gm-removal | include No`	(Optional) Displays only those devices that do not support GM removal.

Removing GMs with Transient IPsec SAs

Perform this task on the KS (or primary KS) to trigger removal of GMs with transient IPsec SAs.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi** [**group** *group-name*] **ks members**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi** [**group** *group-name*] **ks members** **Example:** `Device# clear crypto gdoi ks members`	Creates a new set of TEK and KEK keys. This command also sends out GM removal messages to all GMs to clean up their old TEK and KEK databases.

Examples

A message appears on the KS as follows:

```
Device# clear crypto gdoi ks members

% This GM-Removal message will shorten all GMs' key lifetimes and cause them to
re-register before keys expiry.
Are you sure you want to proceed? ? [yes/no]: yes
Sending GM-Removal message to group GET...
```

After each GM receives the GM removal message, the following syslog message appears on each GM:

```
*Jan 28 08:37:03.103: %GDOI-4-GM_RECV_DELETE: GM received delete-msg from KS in group GET.

TEKs lifetime are reduced and re-registration will start before SA expiry
```

Each GM removes the KEK immediately and shortens the lifetimes of the old TEKs as follows:

```
TEK_SLT = MIN(TEK_RLT, MAX(90s, MIN(5%(TEK_CLT), 3600s)))
TEK_SLT: TEK shortened lifetime
TEK_RLT: TEK Remaining LIfeTime
TEK_CLT: TEK Configured LIfeTime
```

Also, the GMs start re-registering to the KS to obtain the new TEKs and KEK according to the conventional re-registration timer and with jitter (random delay) applied. Jitter prevents all GMs from reregistering at the same time and overloading the key server CPU. Only GMs that pass the authentication based on the new credential installed on the KS will receive the new TEKs and KEK.

GM removal should not cause a network disruption, because traffic continues to be encrypted and decrypted using the transient IPsec SA until its lifetime expires.

If you try to use this command on the secondary KS, it is rejected as follows:

```
Device# clear crypto gdoi ks members

ERROR for group GET: can only execute this command on Primary KS
```

Removing GMs and Deleting IPsec SAs Immediately

Perform this task on the KS (or primary KS) to force GMs to delete old TEKs and KEKs immediately and re-register.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi** [**group** *group-name*] **ks members now**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi** [**group** *group-name*] **ks members now** **Example:** `Device# clear crypto gdoi ks members now`	Creates a new set of TEK and KEK keys. This command also sends out GM removal messages to all GMs to clean up their old TEK and KEK databases. **Note** Using the **now** keyword can cause a network disruption to the data plane. Proceed with the GM removal only if a security concern is more important than a disruption.

Examples

A message appears on the KS as follows:

```
Device# clear crypto gdoi ks members now

% This GM-Removal immediate message will cleanup all GMs downloaded policies
% This will cause all GMs to re-register.
Are you sure you want to proceed? ? [yes/no]: yes
Sending GM-Removal message to group GET...
```

After you enter the above command, the KS sends a “remove now” message to each GM to trigger the following actions on each GM:

1. Immediately cleans up its downloaded TEKs and KEK and its policy and returns to fail-open mode (unless fail-close mode is explicitly configured).
2. Sets up a timer with a randomly chosen period within 2 percent of the configured TEK lifetime.
3. When the timer in Step 2 expires, the GM starts re-registering to the KS to download the new TEKs and KEK.

On each GM, the following syslog message is displayed to indicate that the GM will re-register in a random time period:

```
*Jan 28 08:27:05.627: %GDOI-4-GM_RECV_DELETE_IMMEDIATE: GM receive REMOVAL-NOW in group
GET to cleanup downloaded policy now. Re-registration will start in a randomly chosen
period of 34 sec
```

If you try to remove GMs in a network containing devices that do not support GM removal, a warning message appears:

```
Device# clear crypto gdoi ks members now

% This GM-Removal immediate message will cleanup all GMs downloaded policies
% This will cause all GMs to re-register.
Are you sure you want to proceed? ? [yes/no]: yes
WARNING for group GET: some devices cannot support GM-REMOVAL and can cause network
disruption. Please check 'show crypto gdoi feature'.
Are you sure you want to proceed ? [yes/no]: no
```

Ensuring that GMs Are Running Software Versions That Support Policy Replacement

Perform this task on the KS (or primary KS) to check whether all devices in the network support policy replacement.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature policy-replace**
3. **show crypto gdoi feature policy-replace | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature policy-replace** **Example:** `Device# show crypto gdoi feature policy-replace`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports policy replacement.
Step 3	**show crypto gdoi feature policy-replace | include No** **Example:** `Device# show crypto gdoi feature policy-replace | include No`	(Optional) Finds only those devices that do not support policy replacement. For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs. This behavior is the same as the existing rekey method and ensures backward compatibility.

Triggering a Rekey

If you change the security policy (for example, from DES to AES) on the KS (or primary KS) and exit from global configuration mode, a syslog message appears on the KS indicating that the policy has changed and a rekey is needed. You enter the rekey triggering command as described below to send a rekey based on the latest policy in the running configuration.

Perform this task on the KS (or primary KS) to trigger a rekey.

SUMMARY STEPS

1. **enable**
2. **crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**] **Example:** `Device# crypto gdoi ks group mygroup rekey`	Triggers a rekey on all GMs. The optional **replace-now** keyword immediately replaces the old TEKs and KEK on each GM to enable the new policy before the SAs expire. **Note** Using the **replace-now** keyword could cause a temporary traffic discontinuity.

Examples

A message appears on the KS as follows:

```
Device# crypto gdoi ks rekey
Device#
*Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey with
policy-replace for group GET from address 10.0.8.1 with seq # 2
```

After the policy change, when each GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). Each GM continues to encrypt and decrypt traffic using the old SA until its shortened lifetime expires.

If you try to trigger a rekey on the secondary KS, it rejects the command as shown below:

```
Device# crypto gdoi ks rekey
ERROR for group GET: This command must be executed on Pri-KS
```

Configuration Examples for GET VPN GM Removal and Policy Trigger

Example: Removing GMs from the GET VPN Network

Ensuring That GMs Are Running Software Versions That Support GM Removal

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in the network support the GM removal feature:

```
Device# show crypto gdoi feature gm-removal

Group Name: GET
 Key Server ID       Version   Feature Supported
    10.0.8.1         1.0.2          Yes
    10.0.9.1         1.0.2          Yes
    10.0.10.1        1.0.2          Yes
    10.0.11.1        1.0.2          Yes
 Group Member ID     Version   Feature Supported
    10.0.0.2         1.0.2          Yes
    10.0.0.3         1.0.1          No
```

The following example shows how to find only those devices that do not support GM removal:

```
Device# show crypto gdoi feature gm-removal | include No

       10.0.0.3          1.0.1          No
```

The above example shows that the GM with IP address 10.0.0.3 is running older software version 1.0.1 (which does not support GM removal) and should be upgraded.

Removing GMs with Transient IPsec SAs

The following example shows how to trigger GM removal with transient IPsec SAs. You use this command on the KS (or primary KS).

```
Device# clear crypto gdoi ks members

% This GM-Removal message will shorten all GMs' key lifetimes and cause them to
re-register before keys expiry.
Are you sure you want to proceed? ? [yes/no]: yes
Sending GM-Removal message to group GET...
```

Removing GMs and Deleting IPsec SAs Immediately

The following example shows how to force GMs to delete old TEKs and KEKs immediately and re-register. You use this command on the KS (or primary KS).

```
Device# clear crypto gdoi ks members now

% This GM-Removal immediate message will cleanup all GMs downloaded policies
% This will cause all GMs to re-register.
Are you sure you want to proceed? ? [yes/no]: yes
Sending GM-Removal message to group GET...
```

Example: Triggering Rekeys on Group Members

Ensuring That GMs Are Running Software Versions That Support Rekey Triggering

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to display the version of software on devices in the GET VPN network and display whether they support rekey triggering after a policy change:

```
Device# show crypto gdoi feature policy-replace

Key Server ID       Version  Feature Supported
  10.0.8.1         1.0.2          Yes
  10.0.9.1         1.0.2          Yes
  10.0.10.1        1.0.2          Yes
  10.0.11.1        1.0.2          Yes
 Group Member ID     Version  Feature Supported
  5.0.0.2          1.0.2          Yes
  9.0.0.2          1.0.1          No
```

The following example shows how to find only those devices that do not support rekey triggering after policy replacement:

```
Device# show crypto gdoi feature policy-replace  | include No

       9.0.0.2          1.0.1          No
```

For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs.

Triggering a Rekey

The following example shows how to trigger a rekey after you have performed a policy change. In this example, an IPsec policy change (for example, DES to AES) occurs with the **profile gdoi-p2** command:

```
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# no profile gdoi-p
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# end
```

```
Device#

*Jan 28 09:15:15.527: %SYS-5-CONFIG_I: Configured from console by console
*Jan 28 09:15:15.527: %GDOI-5-POLICY_CHANGE: GDOI group GET policy has changed. Use
'crypto gdoi ks rekey' to send a rekey, or the changes will be send in the next scheduled
rekey
Device# crypto gdoi ks rekey
Device#
*Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey with
policy-replace for group GET from address 10.0.8.1 with seq # 2
```

The following example shows the error message that appears if you try to trigger a rekey on the secondary KS:

```
Device# crypto gdoi ks rekey

ERROR for group GET: This command must be executed on Pri-KS
```

Note If time-based antireplay (TBAR) is set, the key server periodically sends a rekey to the group members every 2 hours (7200 sec). In the following example, even though the lifetime is set to 8 hours (28800 sec), the rekey timer is set to 2 hours.

```
Device(config)# crypto ipsec profile atm-profile
Device(ipsec-profile)# set security-association lifetime seconds 28800
!
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group ATM-DSL
Device(config-gdoi-group)# server local
Device(gdoi-sa-ipsec)# sa ipsec 1
!
Device(gdoi-sa-ipsec)# replay time window-size 100
```

The commands **show crypto gdoi gm replay** and **show crypto gdoi ks replay** displays TBAR information.

Additional References for GET VPN GM Removal and Policy Trigger

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GET VPN GM Removal and Policy Trigger

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 291: Feature Information for GET VPN GM Removal and Policy Trigger

Feature Name	Releases	Feature Information
GET VPN GM Removal and Policy Trigger		This feature provides a command that lets you efficiently eliminate unwanted GMs from the GET VPN network, provides a rekey triggering command to install new SAs and remove obsolete SAs, and provides commands that display whether devices on the network are running versions of GET VPN software that support these features. The following commands were introduced or modified: **clear crypto gdoi, crypto gdoi ks, show crypto gdoi.**

CHAPTER 221

GDOI MIB Support for GET VPN

The existing MIBs in crypto are the Internet Key Exchange (IKE) and IP security (IPsec) MIBs, which are not sufficient for Group Domain of Interpretation (GDOI). The GDOI MIB Support for GET VPN feature adds MIB support for RFC 6407, The Group Domain of Interpretation ; it supports only the objects related to the GDOI MIB IETF standard. You can import the GDOI MIB .my file into an SNMP management station and parse it to retrieve the table objects and hierarchy information.

The GDOI MIB consists of objects and notifications (formerly called traps) that include information about GDOI groups, group menber (GM) and key server (KS) peers, and the policies that are created or downloaded. Only "get" operations are supported for GDOI.

To configure GDOI MIB support for GET VPN, see the "Configuring GDOI MIB Support for GET VPN" section.

Information About GDOI MIB Support for GET VPN

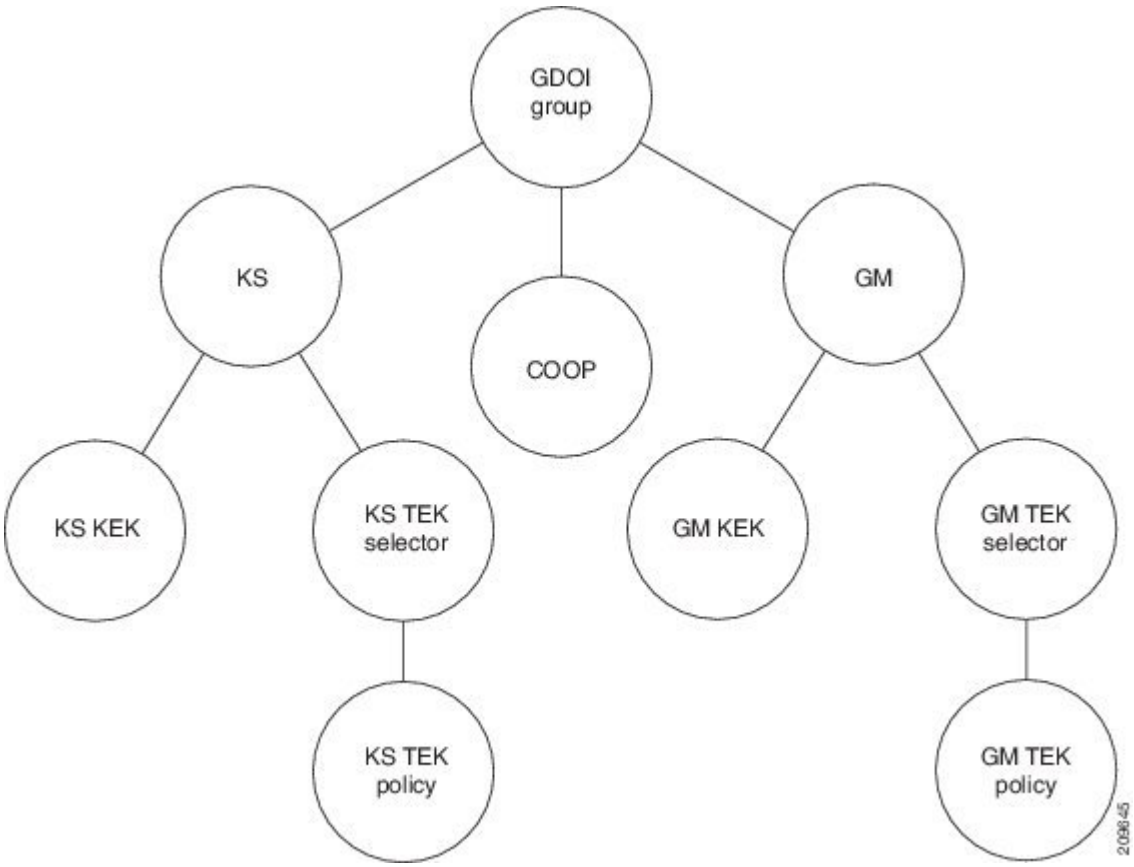
GDOI MIB Compatibility with Other GET VPN Software Versions

The GDOI MIB Support for GET VPN feature provides a command that you use on the KS (or primary KS) to check whether all the devices in the network are running versions that support the GDOI MIB. For more information, see the "Ensuring that GMs Are Running Software Versions That Support the GDOI MIB" section.

GDOI MIB Table Hierarchy

The GDOI MIB objects are organized into the following GDOI MIB tables. Following is the relationship (hierarchy) among the tables:

Figure 126: GDOI MIB Table Hierarchy



GDOI MIB Table Objects

Following is a list of the MIB table objects (listed per group).

Group table objects:

- Group ID type—Specifies whether the group ID is an IP address, group number, hostname, and so on.
- Group ID length—Number of octets in the group ID value.
- Group ID value—Group number, IP address, or hostname.
- Group name—String value.
- Group member count -- Specifies the number of registered KSs to this group.
- Group active peer KS count -- Specifies the number of active KSs to this group.
- Group last rekey retransmits -- Specifies the cumulative count of number of rekey messages and retransmit messages sent as a part of last rekey operation.
- Group last rekey time taken -- Specifies the time taken by the KS to complete the last rekey operation.

KS table objects:

- KS ID type
- KS ID length
- KS ID value
- Active KEK—SPI of the key encryption key (KEK) that is currently used by the KS to encrypt the rekey message.
- Last rekey sequence number—Last rekey number that was sent by the KS to the group.
- KS Role -- Primary or secondary.
- Number of registered GMs -- count of GMs registered to this KS.

COOP table objects:

- COOP peer ID type
- COOP peer ID length
- COOP peer ID value
- COOP peer ID role -- Primary or secondary
- COOP peer status -- Alive, dead or unknown
- Number of registered GMs -- count of GMs registered to the COOP peer

GM table:

- GM ID type
- GM ID length
- GM ID value
- Registered KS ID type—ID type of the KS to which the GM is registered.
- Registered KS ID length
- Registered KS ID value
- Active KEK—SPI of the KEK currently used by the GM to decrypt rekey messages.
- Last rekey seq number—Last rekey number received by the GM.
- Count of active TEKs -- number of active TEKs used by the GM to encrypt/decrypt/authenticate dataplane traffic.

KS KEK table:

- KEK index
- KEK SPI
- KEK source ID information—Source ID type, ID length, and ID value.
- KEK source ID port—Port associated with the source ID.
- KEK destination ID information—Destination ID type, ID length, and ID value.

- KEK destination ID port—Port associated with the destination ID.
- IP protocol ID—UDP or TCP.
- Key management algorithm (unused).
- Encryption algorithm and key length (bits)
- SIG payload hash algorithm, SIG payload signature algorithm, and SIG payload key length (bits).
- Hash algorithm (will be reused from the IPsec MIB)
- Diffie-Hellman group
- KEK original lifetime (seconds)—Maximum time for which a KEK is valid.
- KEK remaining lifetime (seconds)

KS TEK selector table (corresponds to the ACLs that are configured as part of the IPsec SA in the GDOI group configuration on the KS):

- TEK selector index—An integer index for traffic encryption keys (TEK).
- TEK source ID information—Source ID type, ID length, and ID value.
- TEK source ID port—Port associated with the source ID.
- TEK destination ID information—Destination ID type, ID length, and ID value.
- TEK destination ID port—Port associated with the destination ID.
- TEK Security protocol—GDOI_PROTO_IPSEC_ESP protocol ID value in the SA TEK payload (see RFC 6407).

KS TEK policy table:

- TEK policy index—An integer index.
- TEK SPI—Four octets
- Encapsulation mode—Tunnel or transport.
- Encryption algorithm and key length (bits)
- Integrity and authentication algorithm and key length (bits)
- TBAR window size (seconds)
- TEK original lifetime (seconds)—Maximum time for which a TEK is valid.
- TEK remaining lifetime (seconds)
- TEK Status—Inbound, outbound, or not in use.

GM KEK table:

- KEK index—An integer index.
- KEK SPI
- KEK source ID information—Source ID type, ID length , and ID value.

- KEK source ID port—Port associated with the source ID.
- KEK destination ID information—Destination ID type, ID length, and ID value.
- KEK destination ID port—Port associated with the destination ID.
- IP protocol ID—UDP or TCP.
- Key management algorithm (unused)
- Encryption algorithm and key length (bits)
- SIG payload hash algorithm, SIG payload signature algorithm, and SIG payload key length (bits)
- Hash algorithm
- Diffie-Hellman group
- KEK original lifetime (seconds)—Maximum time for which a KEK is valid.
- KEK remaining lifetime (seconds)

GM TEK selector table (corresponds to the ACLs that are downloaded to the GM as part of the TEK policy from the KS):

- TEK selector index—An integer index.
- TEK source ID information—Source ID type, ID length, and ID value.
- TEK source ID port—Port associated with the source ID.
- TEK destination ID information—Destination ID type, ID length , and ID value.
- TEK destination ID port—Port associated with the destination ID.
- TEK Security protocol—GDOI_PROTO_IPSEC_ESP protocol ID value in the SA TEK payload (see RFC 6407).

GM TEK policy table:

- TEK policy index—An integer index.
- TEK SPI —Four octets.
- Encapsulation mode—Tunnel or transport.
- Encryption algorithm and key length (bits)
- Integrity and authentication algorithm and key length (bits)
- TBAR window size (seconds)
- TEK original lifetime (seconds)—Maximum time for which a TEK is valid.
- TEK remaining lifetime (seconds)
- TEK Status—Inbound, outbound, or not in use.

GDOI MIB Notifications

The GDOI MIB supports the Simple Network Management Protocol (SNMP) notifications in the following table. The GDOI MIB contains two kinds of notifications: those generated by the KS and those generated by each GM. You can enable any combination of notifications (or all notifications).

Table 292: SNMP Notifications Supported by the GDOI MIB

Notification	Description
KS New Registration	A KS first received a registration request from a GM.
KS Registration Complete	A GM completed registration to the KS.
KS Rekey Pushed	A rekey message was sent by the KS.
KS No RSA Keys	An error notification was received from the KS because of missing RSA keys.
GM Register	A GM first sent a registration request to a KS.
GM Registration Complete	A GM completed registration to a KS.
GM Re-Register	A GM began the reregistration process with a KS.
GM Rekey Received	A rekey message was received by a GM.
GM Incomplete Config	A GM sent an error notification because of a missing configuration.
GM Rekey Failure	A GM sent an error notification because it cannot process and install a rekey.
KS Role Change	A KS switches between primary and secondary role.
KS GM Deleted	Generated when a GM is deleted from the KS.
KS Peer Reachable	Generated by a KS when unreachable COOP peer becomes reachable.
KS Peer Unreachable	Generated by a KS when reachable COOP peer becomes unreachable.

For more information, see the “Enabling GDOI MIB Notifications” section.

GDOI MIB Limitations

The GDOI MIB contains only objects that are listed in RFC 6407 and does not contain objects for functionality specific to the Cisco implementation of GDOI. This functionality includes:

- Cooperative key servers
- GM ACLs
- Receive-only SAs
- Fail-close/fail-open
- Crypto map objects
- Other Cisco GET VPN-specific features

How to Configure GDOI MIB Support for GET VPN

Ensuring that GMs Are Running Software Versions That Support the GDOI MIB

Perform this task on the KS (or primary KS) to ensure that all devices in the GET VPN network support the GDOI MIB.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature gdoi-mib**
3. **show crypto gdoi feature gdoi-mib | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature gdoi-mib** **Example:** `Device# show crypto gdoi feature gdoi-mib`	Displays the version of the GET VPN software running on each KS and GM in the network and displays whether that device supports the GDOI MIB.
Step 3	**show crypto gdoi feature gdoi-mib | include No** **Example:** `Device# show crypto gdoi feature gdoi-mib | include No`	(Optional) Finds only those devices that do not support the GDOI MIB.

Creating Access Control for an SNMP Community

You specify an SNMP community access string to define the relationship between the SNMP manager and the SNMP agent on the KS or GM in order to permit access to SNMP. Your community access string acts like a password to regulate access to the agent on the device.

Perform this task to specify the community access string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *community-string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*]

4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *community-string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*] **Example:** `Device(config)# snmp-server community mycommunity`	Specifies the community access string.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

For more information about specifying a community access string, refer to the "Configuring SNMP Support" module in the *SNMP Configuration Guide*. For more information about the **snmp-server community** command (including syntax and usage guidelines), refer to the Cisco IOS SNMP Support Command Reference.

Enabling Communication with the SNMP Manager

Perform this task to enable communication between the SNMP agent on the KS or GM and the SNMP manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** {*hostname* | *ip-address*} **version** {**1** | **2c** | **3**} *community-string*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** {*hostname* | *ip-address*} **version** {**1** | **2c** | **3**} *community-string* **Example:** `Device(config)# snmp-server host 209.165.200.225 version 2c mycommunity`	Specifies the host to receive SNMP notifications. • 2c is usually used as the SNMP version.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

For more information about enabling communication with the SNMP manager, refer to the "Configuring SNMP Support" module in the *SNMP Configuration Guide*. For more information about the **snmp-server host** command (including syntax and usage guidelines), refer to the Cisco IOS SNMP Support Command Reference.

Enabling GDOI MIB Notifications

Perform this task to enable GDOI MIB notifications on the KS or GM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps gdoi** [*notification-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**snmp-server enable traps gdoi** [*notification-type*] **Example:** `Device(config)# snmp-server enable traps gdoi gm-registration-complete gm-rekey-rcvd ks-new-registration ks-reg-complete`	Specifies the particular SNMP notifications to be enabled. You can specify any combination of the following types in any order. If you enter the command without any of the following keywords, all GDOI MIB notifications are enabled. • **gm-incomplete-cfg**—A GM sent an error notification because of a missing configuration. • **gm-re-register**—A GM began the reregistration process with a KS. • **gm-registration-complete**—A GM completed registration to a KS. • **gm-rekey-fail**—A GM sent an error notification because it cannot successfully process and install a rekey. • **gm-rekey-rcvd**—A rekey message was received by a GM. • **gm-start-registration**—A GM first sent a registration request to a KS. • **ks-new-registration**—A KS first received a registration request from a GM. • **ks-no-rsa-keys**—An error notification was received from the KS because of missing RSA keys. • **ks-reg-complete**—A GM completed registration to the KS. • **ks-rekey-pushed**—A rekey message was sent by the KS. • **ks-gm-deleted**—A GM is deleted by the KS. • **ks-peer-reachable**—An unreachable COOP peer becomes reachable. • **ks-peer-unreachable**—A reachable COOP peer becomes unreachable. • **ks-role-change**—A KS changes it's role from primary to secondary or vice-versa.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

Configuration Examples for GDOI MIB Support for GET VPN

Example: Ensuring That GMs Are Running Software Versions That Support the GDOI MIB

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in the network support the GDOI MIB:

```
Device# show crypto gdoi feature gdoi-mib

Group Name: GET
 Key Server ID       Version   Feature Supported
    10.0.8.1         1.0.2          Yes
    10.0.9.1         1.0.2          Yes
    10.0.10.1        1.0.2          Yes
    10.0.11.1        1.0.2          Yes
 Group Member ID     Version   Feature Supported
    10.0.11.2          1.0.2          Yes
    10.0.11.3          1.0.1          No
```

The following example shows how to find only those devices that do not support the GDOI MIB:

```
Device# show crypto gdoi feature gdoi-mib | include No

       10.0.11.3          1.0.1          No
```

Example: Creating Access Control for an SNMP Community

The following example shows how to specify an SNMP community string named mycommunity to define the relationship between the SNMP manager and the SNMP agent on the KS or GM in order to permit access to SNMP:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server community mycommunity
Device(config)# end
```

Example: Enabling Communication with the SNMP Manager

The following example shows how to enable communication with the SNMP manager. This example using a community string named mycommunity that has already been created:

```
Device> enable
Device# configure terminal
```

```
Device(config)# snmp-server host 209.165.200.225 version 2c mycommunity
Device(config)# end
```

Example: Enabling GDOI MIB Notifications

The following example shows how to enable GDOI MIB notifications:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server enable traps gdoi gm-registration-complete gm-rekey-rcvd
ks-new-registration ks-reg-complete
Device(config)# end
```

Additional References for GDOI MIB Support for GET VPN

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Configuring SNMP	• "Configuring SNMP Support" module in the SNMP Configuration Guide, Cisco IOS Release 15.2M&T • Cisco IOS SNMP Support Command Reference

MIBs

MIB	MIBs Link
CISCO-GDOI-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GDOI MIB Support for GET VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 293: Feature Information for GDOI MIB Support for GET VPN

Feature Name	Releases	Feature Information
GDOI MIB Support for GET VPN		This feature adds MIB support for IETF RFC 6407, The Group Domain of Interpretation. This feature supports only the objects related to the GDOI MIB IETF standard. This feature also provides a command that displays whether devices on the network are running versions of GET VPN software that support the GDOI MIB. The GDOI MIB consists of objects and notifications that include information about GDOI groups, GM and KS peers, as well as the policies that are created or downloaded. The following command was introduced: **snmp-server enable traps gdoi.**
XE 3.16 GETVPN GDOI/COOP MIBS		The following command was modified: **snmp-server enable traps gdoi.**

CHAPTER 222

GET VPN Resiliency

The GET VPN Resiliency feature improves the resiliency of Cisco Group Encrypted Transport (GET) VPN so that data traffic disruption is prevented or minimized when errors occur.

Prerequisites for GET VPN Resiliency

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.4 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support this feature. For more information, see the "*Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime*" section.

Restrictions for GET VPN Resiliency

- All key servers (KSs) and group members (GMs) must be upgraded for Long SA Lifetime.

Information About GET VPN Resiliency

Long SA Lifetime

The long security association (SA) lifetime functionality extends the maximum lifetime of the key encryption key (KEK) and traffic encryption key (TEK) from 24 hours to 30 days. This functionality also lets you configure key servers (KSs) to continue to send periodic reminder rekeys to group members (GMs) that do not respond with an acknowledgment in the last scheduled rekey.

By using a long SA lifetime in combination with periodic reminder rekeys, a KS can effectively synchronize GMs if they miss a scheduled rekey before the keys roll over.

Note For a lifetime longer than 24 hours, the encryption algorithm must be Advanced Encryption Standard-cipher block chaining (AES-CBC) or Advanced Encryption Standard-Galois/Counter Mode (AES-GCM) with an AES key of 128 bits or stronger.

You can use the long SA lifetime functionality along with the GETVPN Suite-B feature to use AES-GSM and Galois Message Authentication Code-Advanced Encryption Standard (GMAC-AES) as traffic encryption key (TEK) policy transforms in a group for packets encapsulated with GCM-AES and GMAC-AES.

Migrating to Long SA Lifetime

When migrating to the long SA lifetime functionality (greater than or equal to one day), the following rules apply:

- When a long SA lifetime is configured on a crypto IPsec profile, GETVPN displays a warning message to not use the IPsec profile for a non- Group Domain of Interpretation (GDOI) group.
- If group members are registered to a key server with short SA lifetime and the key server changes the policy to long SA lifetime, GETVPN checks the software version of all the GMs when the **crypto gdoi ks rekey** command is configured to initiate the policy change. If the GMs registered with the KS do not support long SA lifetime, a message is displayed to discourage the policy change until all GMs are upgraded.
- When the Long SA feature is enabled in KS, it will block registration from GMs running older Cisco IOS releases, which does not support this feature.

Clock Skew Mitigation

Sometimes with longer security association (SA) lifetimes, a group member (GM) may not receive updates from a key server for a longer duration. This may result in group members experiencing clock skew for key encryption key (KEK) lifetime, traffic encryption key (TEK) lifetime, and Time-Based Anti-Replay (TBAR) pseudotime. The refresh rekey and rollover to new outbound IPsec SA helps GMs in mitigating clock skew issues.

Refresh Rekey

If the traffic encryption key (TEK) lifetime is set for a duration greater than two days and Time-Based Anti-Replay (TBAR) is disabled, a key server sends a refresh rekey every 24 hours which updates the key encryption key (KEK) lifetime, TEK lifetime, and TBAR pseudotime on all group members (GMs). In simple terms, a refresh rekey is a retransmission of the current KEK policy, TEK policy, and TBAR pseudotime (if enabled) to all GMs, regardless of the status of receiving a unicast acknowledgment (ACK) for the last rekey. If TBAR is enabled, the refresh rekey is sent every two hours to synchronize the pseudotime, so that an additional refresh rekey is not required.

Rollover to New Outbound IPsec SA

When a long SA lifetime (greater than one day) is configured, the rollover happens when the remaining lifetime of the traffic encryption key (TEK) reaches 1% of the old TEK configured lifetime that has a lower limit of 30 seconds and not 30 seconds of the old TEK's remaining lifetime. This allows a greater clock skew between the group members (GMs) before discarding traffic from one GM rolling over to the new TEK late (after the

other GM has already deleted the old TEK). This mitigates the GM from being “offline” (disconnected from the KS) for a long duration and from being unable to receive the refresh rekeys to mitigate the clock skew.

Periodic Reminder Sync-Up Rekey

The periodic reminder sync-up rekey functionality in the key server (KS) lets you to send periodic reminder rekeys to group members (GMs) who do not respond with an acknowledgment (ACK) in the last scheduled rekey. This functionality in combination with the long SA lifetime functionality is effective for a KS to synchronize with GMs when they miss a scheduled rekey before the keys rollover. In a KS group configuration, a new keyword **periodic** is added to the **rekey retransmit** command when configuring the rekey retransmission.

Each periodic rekey increments the sequence number, similar to rekey retransmissions. The GM is removed from the database on the KS after 3 scheduled rekeys (not retransmissions) for which the GM does not send an ACK.

Pre-Positioned Rekey

The pre-positioned rekey functionality allows the key server (KS) to send a rekey earlier than half the duration of the SA lifetime, when a longer SA lifetime (greater than one day) is configured. The normal behavior of sending the rekey is used for a short SA lifetime. When group members (GMs) receive this early rekey, they continue to use the old TEK as outbound until rolled over to the new TEK as outbound. The pre-positioned rekey feature along with the Long SA Lifetime feature improves key rollover stability. This functionality allows the (KS) sufficient time to recover rekey errors, such as periodic reminder rekeys and synchronize rekeys.

How to Configure GET VPN Resiliency

Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime

You should use the Long SA Lifetime feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature.

Perform this task on the key server (or primary key server) to ensure that all devices in the network support long SA lifetime.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature long-sa-lifetime**
3. **show crypto gdoi feature long-sa-lifetime | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**show crypto gdoi feature long-sa-lifetime** **Example:** `Device# show crypto gdoi feature long-sa-lifetime`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports long SA lifetime.
Step 3	**show crypto gdoi feature long-sa-lifetime | include No** **Example:** `Device# show crypto gdoi feature long-sa-lifetime | include No`	(Optional) Displays only those devices that do not support long SA lifetime.

Configuring Long SA Lifetime

Configuring Long SA Lifetime for TEK

To configure long SA lifetime for traffic encryption key (TEK), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*
4. **set security-association lifetime days** *days*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Device(config)# crypto ipsec profile gdoi-p`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters crypto IPsec profile configuration mode.
Step 4	**set security-association lifetime days** *days* **Example:** `Device(ipsec-profile)# set security-association lifetime days 15`	Configures the security association (SA) lifetime to over one day. • The maximum number of days is 30.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(ipsec-profile)# end`	Exits crypto IPsec profile configuration mode and returns to privileged EXEC mode.

Configuring Long SA Lifetime for KEK

To configure long SA lifetime for key encryption key (TEK), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *number*
5. **server local**
6. **rekey lifetime days** *days*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GET`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 3333`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey lifetime days** *days* **Example:** `Device(gdoi-local-server)# rekey lifetime days 20`	Limits the number of days or seconds for a KEK.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring the Periodic Reminder Sync-Up Rekey

To configure the periodic reminder sync-up rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *number*
5. **server local**
6. **rekey retransmit** *number-of-seconds* **periodic**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group group1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 3333`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.

	Command or Action	Purpose
Step 6	**rekey retransmit** *number-of-seconds* **periodic** **Example:** `Device(gdoi-local-server)# rekey retransmit 10 periodic`	Specifies the number of times the rekey message is periodically retransmitted. • If this command is not configured, there will be no retransmits.
Step 7	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Verifying and Troubleshooting GET VPN Resiliency

Verifying and Troubleshooting GET VPN Resiliency on a Key Server

To view the configuration that is running on a key server (KS), use the **show running-config** command and the following commands.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi**
3. **show crypto gdoi ks rekey**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi** **Example:** `Device# show crypto gdoi`	Displays the current GDOI configuration and the policy that is downloaded from the KS.
Step 3	**show crypto gdoi ks rekey** **Example:** `Device#  show crypto gdoi ks rekey`	Displays information about the rekeys that are sent from the KS.

Verifying and Troubleshooting GET VPN Resiliency on a Group Member

To view the configuration that is running on a group member (GM), use the **show running-config** command and the following commands.

SUMMARY STEPS

1. **enable**

2. **show crypto gdoi ks rekey**
3. **show crypto gdoi ks policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi ks rekey** **Example:** `Device#  show crypto gdoi ks rekey`	Displays information about the rekeys that are sent from the KS.
Step 3	**show crypto gdoi ks policy** **Example:** `Device# show crypto gdoi ks policy`	Displays the time until the next rekey.

Configuration Examples for GET VPN Resiliency

Example: Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support long SA lifetimes:

```
Device# show crypto gdoi feature long-sa-lifetime

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.4          Yes
        10.0.6.2         1.0.4          Yes
        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No

    Group Member ID     Version   Feature Supported
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
        10.0.3.1         1.0.4          Yes
        10.0.3.2         1.0.4          Yes
```

You can also enter the above command on a GM (which will display the information for the GM but not for the KS or other GMs).

The following example shows how to enter the command on the KS (or primary KS) find only those devices in the GET VPN network that do *not* support long SA lifetimes:

```
Device# show crypto gdoi feature long-sa-lifetime | include No

        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
```

Example: Configuring Long SA Lifetime

Example: Configuring Long SA Lifetime for TEK

The following example shows how to configure the long SA lifetime for traffic encryption key (TEK):

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec profile gdoi-p
Device(ipsec-profile)# set security-association lifetime days 15
Device(ipsec-profile)# end
```

Example: Configuring Long SA Lifetime for KEK

The following example shows how to configure the long SA lifetime for key encryption key (KEK):

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey lifetime days 20
Device(gdoi-local-server)# end
```

Example: Configuring the Periodic Reminder Sync-Up Rekey

The following example shows how to configure the periodic reminder sync-up rekey:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group group1
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey retransmit 10 periodic
Device(gdoi-local-server)# end
```

Additional References for GET VPN Resiliency

Related Documents

Related Topic	Document Title

Related Topic	Document Title
Cisco IOS security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GET VPN Resiliency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 294: Feature Information for GET VPN Resiliency

Feature Name	Releases	Feature Information
GET VPN Resiliency		The following commands were introduced or modified: **rekey lifetime, rekey retransmit, set security-association lifetime, show crypto gdoi.**

CHAPTER 223

GETVPN Resiliency GM - Error Detection

The GETVPN Resiliency - GM Error Detection feature detects erroneous packets in the data plane for each Group Domain of Interpretation (GDOI) group such as invalid stateful packet inspections (SPIs) or Time-Based Anti-Replay (TBAR) errors. These errors are tracked, and the outer source IP address of the packet is recorded.

Information About GETVPN Resiliency - GM Error Detection

Error Handling

The GETVPN Resiliency - GM Error Detection feature should be enabled on both the GM and KS for error handling to work. The KS encodes the group information in the SPI (Security Parameter Index) and then it downloads it via the TEK policy to the GM.

When a failure is detected by the GETVPN Resiliency - GM Error Detection feature, a syslog message is generated to show the source IP address of the erroneous packet:

```
 *Feb 10 21:01:56.043:
%GDOI-4-TIMEBASED_REPLAY_FAILED: An anti replay check has failed in
group GETVPN from sourceip-address
100.0.0.9.
 my_pseudotime is 600006.78 secs,
peer_pseudotime is 500033.34 secs, replay_window is 100
(second)
 *Feb 10 21:01:56.043:
%CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed
 connection id=29, sequence
number=11
```

The **show crypto gdoi gm** command displays the history of the last 50 Time-Based Anti-Replay (TBAR) errors. You can use these source IP address records to track down the sender group members (GMs) and investigate any existing hardware or software problems. The following statistical information is also available in the command:

- GM recovery feature ON/OFF

- Interval between recoveries
- Number of GM recovery reregistration enforced

When errors occur, the GM reregisters to the next available key server (KS) to retrieve the latest policy and keys and maintains all previously downloaded group policies and keys until the registration is complete.

For instance, when a cooperative key server (COOP KS) split occurs, each promoted KS generates its own Key Encryption Key (KEK) and Traffic Encryption Key (TEK). When a GM receives invalid SPI packets, it will decode it (the KS encodes the group information in the SPI and then it downloads it via the TEK policy to the GM) and if it finds that it belongs to the current getvpn group then it will start the recovery registration.

An invalid SPIs can belong to one of the following two categories:

- Positive invalid SPI: An invalid SPI that belong to the current group and require GM recovery registration.
- Negative invalid SPI: An invalid SPI that does not require recovery registration.

In the case of a positive invalid SPI, a recovery registration to the next key server (KS) on its list is performed. This recovery registration is repeated for each invalid stateful packet inspection (SPI) packet or TBAR error in each client recovery interval to the next KS on the list. When all the KSs in the list are recovered and no longer contain the invalid SPI, that SPI is marked as a false positive and no more recovery registrations are performed. The KSs will always do the recovery registration for TBAR errors. However, once the GM recovers to all the KSs in the list because of an invalid SPI and none of the KSs has that SPI, it will mark that SPI as a false positive and will not do more recovery registrations due to that SPI.

A syslog message is generated to notify you that this GM recovery reregistration feature is triggered. For instance, if you configure the GM to monitor for control-plane errors every 300 seconds, when the recovery registration occurs the following syslog is generated:

```
*Feb 23 19:06:28.600: %GDOI-5-GM_RECOVERY_REGISTER: received invalid GDOI packets; register
 to KS to refresh policy, keys, and PST.
```

How to Configure GETVPN Resiliency - GM Error Detection

Configuring GETVPN Resiliency - GM Error Detection

SUMMARY STEPS

1. **crypto gdoi group** *group-name*
2. **identity number** *number*
3. **server address ipv4** *address*
4. **client recovery-check interval** *interval*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto gdoi group** *group-name* **Example:**	Creates a Group Domain of Interpretation (GDOI) group and enters GDOI group configuration mode.

	Command or Action	Purpose
	`Device(config)# crypto gdoi group GETVPN`	
Step 2	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 1111`	Identifies a GDOI group number.
Step 3	**server address ipv4** *address* **Example:** `Device(config-gdoi-group)# server address ipv4 1.0.0.2`	Specifies the IP address of the server that the GDOI group is trying to reach.
Step 4	**client recovery-check interval** *interval* **Example:** `Device(config-gdoi-group)# client recovery-check interval 300`	Sets the interval of time for the client group member (GM) to monitor for control-plane errors.
Step 5	**exit** **Example:** `Device(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.

Configuration Examples for GETVPN Resiliency - GM Error Detection

Example: Configuring GETVPN Resiliency - GM Error Detection

The following example shows how to enable the group member (GM) to monitor for control-plane errors every 300 seconds.

```
crypto gdoi group GETVPN
 identity number 1111
 server address ipv4 1.0.0.2
 client recovery-check interval 300
```

Additional References for GETVPN Resiliency - GM Error Detection

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GETVPN Resiliency - GM Error Detection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 295: Feature Information for GETVPN Resiliency - GM Error Detection

Feature Name	Releases	Feature Information
GETVPN Resiliency - GM Error Detection		Detects erroneous packets in the data plane for each GDOI group. The following command was introduced: **client recovery-check interval.**

CHAPTER 224

GETVPN CRL Checking

During the Group Encrypted Transport VPN (GET VPN) process, certificates are received from a certificate authority (CA) and used as a proof of identity. Certificates may be revoked for a number of reasons, such as key compromise or certificate loss. Revoked certificates are placed on a certificate revocation list (CRL) that is published periodically to a repository. This list is stored on the repository for the length of time specified by a configured CRL lifetime, and can be anything from a few hours to several days.

Information About GETVPN CRL Checking

In Internet Key Exchange (IKE), certificates are validated when a session is established between two peers. Current sessions are not affected by certificate revocation. However, new sessions will fail to establish and certificates are not validated again unless group members reregister to the key server (KS).

The GETVPN CRL Checking feature enables public key infrastructure (PKI) to notify Group Domain of Interpretation (GDOI) KSs when a new CRL is available for a configured trustpoint. The KS then creates a new Key Encryption Key (KEK) and sends a reauthentication message to the group member devices, which print a syslog message, delete the current KEKs, and reregister to the KS.

Cooperative Key Server Protocol Integration

Cooperative Key Server Protocol (COOP) is a feature of GET VPN that allows you to configure multiple key servers (KSs) in a VPN network. It is used for KS redundancy.

GETVPN CRL checking integrates with COOP by enabling group member (GM) reauthentication on all KSs. However there is always a possibility that a COOP split may occur, where connectivity is temporarily lost among cooperative KSs.

No COOP Split when Reauthentication is Triggered

If no COOP split occurs the primary GM device deletes the Key Encryption Key (KEK) to secondary KSs and sends a reauthentication message to GMs. The secondary KSs then have the current policies synchronized

with the primary policies before the GMs start to reregister. All GMs reregister and reauthenticate to an available KS and receive the new KEK.

COOP Split when Reauthentication is Triggered

If a COOP split occurs before reauthentication is triggered and there are only two primary KSs, they both send out the reauthentication message. Each primary KS creates a new and different KEK. The GM only understands the first reauthentication message it receives as it deletes all the existing KEKs immediately after receiving the message. The GM then reregisters to an available KS and a CRL check takes place. When reregistering, the GM receives either the KEK of the first primary or the KEK of the second primary, depending on which KS the GM reregistered. The GM then installs that KEK and receives further rekeys only from that primary KS. When the COOP merge occurs, the KSs sync up the policies and send rekeys so that all GMs have the current KEK and traffic encryption keys (TEKs).

Avoiding the Creation of Different KEKs

Reauthentication and CRL checking still occurs if reauthentication is triggered during a COOP split. However, triggering the creation of different KEKs in the KSs is avoided by delaying reauthentication. A primary KS only starts the reauthentication if all COOP KSs are reachable (not split). If one COOP KS is not reachable, the primary KS delays sending the reauthentication message until all COOP KSs are reachable.

How to Configure GETVPN CRL Checking

You need to configure several components prior to enabling the GETVPN CRL Checking feature. These include:

- A defined public key infrastructure (PKI) certificate authority (CA) so that group members and key servers are PKI clients and, therefore must enroll to get certificates.
- Key servers (KSs) configured to have certificate revocation list (CRL) checking enabled in PKI.
- KSs configured to download the CRL when it is available on the CA and on a first-needed basis. This means that the KSs download the CRL following the first group member (GM) registration after the new CRL is available. See the “Configuring Key Servers for GETVPN CRL Checking” section.
- CRL checking disabled on the group member devices for PKI. See the “Disabling CRL Checking on Group Members” section.
- Internet Key Exchange (IKE) authentication set to certificates. See the “Setting IKE Authentication to Certificates” section

Configuring Key Servers for GETVPN CRL Checking

To configure key servers (KSs) to download the certificate revocation list (CRL) when the first group member (GM) registration occurs after a new CRL is available on the certificate authority (CA), perform the following steps:

SUMMARY STEPS

1. **ip domain name** *name*
2. **ip http server**

3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **revocation-check** *method*
6. **exit**
7. **crypto identity** *method*
8. **fqdn** *domain*
9. **fqdn** *domain*
10. **exit**
11. **crypto gdoi group** *group-name*
12. **server local**
13. **authorization identity** *name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip domain name** *name* **Example:** `Device(config)# ip domain name cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 2	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on an IP or IPv6 system.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint mycert`	Defines the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Device(config-ca-trustpoint)# enrollment url http://10.1.3.1:80`	Specifies the enrollment URL of the CA.
Step 5	**revocation-check** *method* **Example:** `Device(config-ca-trustpoint)# revocation-check crl`	Ensures certificate checking is performed by a CRL.
Step 6	**exit** **Example:** `Device(config-ca-trustpoint)# exit`	Exits CA trustpoint configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**crypto identity** *method* **Example:** `Device(config)# crypto identity abcd`	Configures the identity of the device with a given list of distinguished names (DNs) in the certificate of the device and enters crypto identity configuration mode. **Note** You can set restrictions in the device configuration that prevent peers with specific certificates, especially certificates with particular DNs, from having access to selected encrypted interfaces.
Step 8	**fqdn** *domain* **Example:** `Device(config-crypto-identity)# fqdn ut01-unix5.cisco.com`	Derives the name mangler from the remote identity of the fully qualified domain name (FQDN) for a GM.
Step 9	**fqdn** *domain* **Example:** `Device(config-crypto-identity)# fqdn ut01-unix6.cisco.com`	Derives the name mangler from the remote identity of the FQDN for the next GM.
Step 10	**exit** **Example:** `Device(config-crypto-identity)# exit`	Exits crypto identity configuration mode and returns to global configuration mode.
Step 11	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoi-group1`	Creates a Group Domain of Interpretation (GDOI) group and enters GDOI group configuration mode.
Step 12	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 13	**authorization identity** *name* **Example:** `Device(config-gdoi-local-server)# authorization identity abcd`	Specifies an authorization identity for a GDOI group based on a distinguished name (DN) or FQDN,
Step 14	**end** **Example:** `Device(config-gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Disabling CRL Checking on Group Members

To disable certificate revocation list (CRL) checking on group members (GMs) for public key infrastructure (PKI), perform the following steps:

SUMMARY STEPS

1. **ip domain name** *name*
2. **ip http server**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **revocation-check** *method*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip domain name** *name* **Example:** `Device(config)# ip domain name cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 2	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on an IP or IPv6 system.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint mycert`	Defines the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Device(config-ca-trustpoint)# enrollment url http://10.1.3.1:80`	Specifies the enrollment URL of the certificate authority (CA).
Step 5	**revocation-check** *method* **Example:** `Device(config-ca-trustpoint)# revocation-check none`	Disables certificate checking on the GMs.
Step 6	**exit** **Example:** `Device(config-ca-trustpoint)# exit`	Exits CA trustpoint mode and returns to global configuration mode.

Setting IKE Authentication to Certificates

SUMMARY STEPS

1. **crypto isakmp policy** *priority*
2. **no authentication pre-share**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Defines an internet key exchange (IKE) policy and enters ISAKMP policy configuration mode.
Step 2	**no authentication pre-share** **Example:** `Router(config-isakmp)# no authentication pre-share`	Resets the authentication method within the IKE policy to the default value.
Step 3	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling GETVPN CRL Checking on Key Servers

To configure public key infrastructure (PKI) to notify the Group Domain of Interpretation (GDOI) key server (KS) when a new certificate revocation list (CRL) is available for the configured trustpoint certificate authority (CA), perform the following steps:

SUMMARY STEPS

1. **crypto gdoi group** *group-name*
2. **server local**
3. **registration periodic crl trustpoint** *trustpoint-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoi_group1`	Creates a GDOI group and enters GDOI group configuration mode.

	Command or Action	Purpose
Step 2	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 3	**registration periodic crl trustpoint** *trustpoint-name* **Example:** `Device(config-gdoi-local-server)# registration periodic crl trustpoint mycert`	Enables periodic registrations for the GDOI KSs when new CRLs become available for the configured PKI trustpoint certificate authority.
Step 4	**end** **Example:** `Device(config-gdoi-local-server)# end`	Exits GDOI local server mode and returns to privileged EXEC mode.

Configuration Examples for GETVPN CRL Checking

Example: Enabling GETVPN CRL Checking

The following examples show how the GETVPN CRL checking feature is enabled, including all required preconfigurations.

Example: Configuring Key Servers for GETVPN CRL Checking

In the following example, thekey servers (KSs)are configured to download the certificate revocation list (CRL) when the first group member registration occurs after a new CRL is available on the trustpoint certificate authority (CA) named mycert:

```
ip domain name cisco.com
ip http server
crypto pki trustpoint mycert
 enrollment url http://10.1.3.1:80
 revocation-check crl

crypto identity abcd
 fqdn ut01-unix5.cisco.com
 fqdn ut01-unix6.cisco.com

crypto gdoi group gdoi-group1
 server local
 authorization identity abcd
```

Example: Disabling CRL Checking on Group Members

In the following example, CRL checking on Group Members (GM) for public key infrastructure (PKI) is disabled:

```
ip domain name cisco.com
ip http server
crypto pki trustpoint mycert
 enrollment url http://10.1.3.1:80
 revocation-check none
```

Example: Setting IKE Authentication to Certificates

```
crypto isakmp policy 1
 no authentication pre-share
```

Example: Enabling GETVPN CRL Checking on Key Servers

In the following example, PKI is configured to notify the GDOI KS named group1 when a new CRL is available for the trustpoint CA named mycert:

```
Crypto gdoi group gdoi_group1
 Server local
  registration periodic crl trustpoint mycert
```

Additional References for GETVPN CRL Checking

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GETVPN Solution Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GETVPN CRL Checking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 296: Feature Information for GETVPN CRL Checking

Feature Name	Releases	Feature Information
GETVPN CRL Checking		Enables public key infrastructure (PKI) to notify Group Domain of Interpretation (GDOI) key servers (KSs) when a new certificate revocation list (CRL) is available for a configured trustpoint. The following command was introduced: **registration periodic crl trustpoint.**

CHAPTER 225

GET VPN Support with Suite B

The GET VPN Support with Suite B feature adds support of the Suite B set of ciphers to Cisco Group Encrypted Transport (GET) VPN. Suite B is a set of cryptographic algorithms that includes Galois Counter Mode Advanced Encryption Standard (GCM-AES) as well as algorithms for hashing, digital signatures, and key exchange.

Suite B for IP security (IPsec) VPNs is a standard whose usage is defined in RFC 4869, Suite B Cryptographic Suites for IPsec . Suite B provides a comprehensive security enhancement for Cisco IPsec VPNs, and it allows additional security for large-scale deployments. Suite B is the recommended solution for organizations requiring advanced encryption security for the wide-area network (WAN) between remote sites.

Prerequisites for GET VPN Support with Suite B

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.4 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support Suite B. For more information, see the "Ensuring That GMs Are Running Software Versions That Support Suite B" section.

Restrictions for GET VPN Support with Suite B

When they are using a GCM policy or a Galois Message Authentication Code (GMAC) traffic encryption key (TEK) policy, all cooperative KSs for a group must use an access control list (ACL) that has identical ACL entries (ACEs) in the identical order. If not, GMs that register to separate KSs cannot encrypt and decrypt correctly after downloading the policy. This is because with Suite B, an SPI (security parameter index ID that is associated with the TEK) is generated for *each* ACL entry and is unique to each ACL entry.

You cannot reorder entries in an existing ACL. So if you are using a GCM or GMAC TEK policy and must update the ACL on each KS so that it has identical entries in the identical order on each KS, you must remove the ACL from each secondary KS, then create a new ACL on the primary KS, then copy it to the secondary KSs, and then enter the **crypto gdoi ks rekey** command on the primary KS to trigger a rekey across the GET VPN network.

You remove an ACL by using the **no** form of the **ip access-list** command (or if you are using IPv6, the **no** form of the **ipv6 access-list** command).

Cisco Catalyst 8000 Series Edge Platforms do not support Suite B in GET VPN. Suite B is supported only on the following Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4000 Series Integrated Services Routers.

Table 297: GET VEPN Suite B Support

Platforms	Models	GET VPN Suite B
Cisco ASR 1000 Series Aggregation Services Routers	ASR1001-X	Supported
	ASR1002-X	Supported
	ASR1001-HX	Supported
	ASR1002-HX	Supported
	ESP100	Supported
	ESP200	Supported
Cisco 4000 Series Integrated Services Routers	ISR 4461	Supported
	ISR4451-X	Supported
	ISR4431	Supported

Information About GET VPN Support with Suite B

Suite B

Suite B is standardized by the National Security Agency (NSA) and the National Institute of Standards and Technology (NIST). The GET VPN Support with Suite B feature allows these cryptographic algorithms to be used with GDOI and GET VPN in various ways, including the use of SHA-2/HMAC-SHA-2 and AEC-GCM/AES-GMAC.

Secure Hash Algorithm 2 (SHA-2) is a set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, and SHA-512) designed by the NSA and published by the NIST as a U.S. Federal Information Processing Standard (FIPS). SHA-2 includes many changes from its predecessor, SHA-1. SHA-2 comprises a set of four hash functions with digests that are 224, 256, 384, or 512 bits.

HMAC is a mechanism for message authentication using iterative cryptographic hash functions. HMAC-SHA-2 is HMAC used in combination with the SHA-2 version (SHA-224, SHA-256, SHA-384, and SHA-512) iterative cryptographic hash functions in combination with a secret shared key in IPsec. These combinations

are called HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. These algorithms can be used as the basis for data origin authentication and integrity verification mechanisms for the authentication header (AH) (although not supported by GET VPN), encapsulating security payload (ESP), IKE, and IKEv2 protocols, and also as pseudo-random functions (PRFs) for IKE and IKEv2.

AES using GCM (AES-GCM) is an encryption algorithm for IPsec. AES using Galois Message Authentication Code (AES-GMAC) is a message integrity algorithm also used for IPsec.

SHA-2 and HMAC-SHA-2

The GET VPN Support with Suite B feature lets you use SHA-2 and HMAC-SHA-2 (HMAC-SHA-256, 384, and 512) as the hash and signature algorithms. SHA-2 and HMAC-SHA-2 with 256, 384, & 512-bit keys are used in

- GDOI registration using IKEv1 as the hash algorithm as described in Section 3.2 (authentication between KSs and GMs) of RFC 6407, The Group Domain of Interpretation .
- The key encryption key (KEK) rekey policy to hash the rekey message for authentication of the rekey message from the KS as well as authentication of the acknowledgment message from the GM.
- The TEK IPsec policy as HMAC-SHA-2 for IPsec SA integrity checking.

AES-GCM and AEC-GMAC

AES-GCM (AES-GCM-128, 192, and 256) and AES-GMAC (AES-GMAC-128, 192, and 256) cryptographic algorithms with 256, 384, and 512-bit keys are used in TEK IPsec policies as IPsec SA encryption and integrity algorithms. GCM is used for encryption and integrity, while GMAC is used for integrity only.

Sets of Cryptographic Algorithms that Comply with Suite B

RFC 4869 describes four sets of cryptographic algorithms for use with IKE and IPsec. When configured, any of these sets will comply with Suite B. Each set consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm:

- Suite-B-GCM-128: Provides ESP integrity protection and confidentiality using 128-bit AES-GCM (see RFC 4106, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)). Use this suite or Suite-B-GCM-256 when ESP integrity protection and encryption are both needed.
- Suite-B-GCM-256: Provides ESP integrity protection and confidentiality using 256-bit AES-GCM (see RFC 4106, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)). Use this suite or Suite-B-GCM-128 when ESP integrity protection and encryption are both needed.
- Suite-B-GMAC-128: Provides ESP integrity protection using 128-bit AES-GMAC (see RFC 4543, The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH) but does not provide confidentiality. Use this suite or Suite-B-GMAC-256 only when there is no need for ESP encryption.
- Suite-B-GMAC-256: Provides ESP integrity protection using 256-bit AES-GMAC (see RFC 4543, The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH) but does not provide confidentiality. Use this suite or Suite-B-GMAC-128 only when there is no need for ESP encryption.

Cisco software contains the ability to configure any of these algorithms. The GET VPN Support with Suite B feature allows GET VPN to use these algorithms.

SID Management

In GET VPN, a counter-based mode of operation (for example, ESP-GCM-AES) requires that an initialization vector (IV) is never reused with a group key. Therefore, this feature provides a method to allow a KS to allocate to each GM (for each interface) a unique sender identifier (SID) for IV construction.

In Suite B, TEK IPsec policies that are used as IPsec SA encryption and integrity algorithms require management of unique pools of SID values on KSs to distribute those unique SID values (GMSIDs) to GMs. Each cooperative KS must have a distinct pool of GMSIDs to allocate. Each KS configures unique KS SIDs (KSSIDs) to configure these SID pools.

A SID space is divided into two parts: a KSSID part and a GMSID part. Therefore, a SID is a concatenation of a KSSID and a GMSID, where the KSSID is the KS portion of a SID, and the GMSID is the GM portion of the SID. A SID is formed by the following bits:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3  (bits)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    KSSID    |            GMSID                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In this example, each KSSID (0 to 127) has 2^{17} (131,072) GMSIDs, which are dynamically assigned to each registering GM.

A GM uses GMSIDs to form a unique 64-bit IV for each packet sent with a given key when using AES-GCM or AES-GMAC. An IV is formed by the following bytes:

```
  0   1   2   3   4   5   6   7   (bytes)
+---+---+---+---+---+---+---+---+
|    SID    |       SSIV        |
+---+---+---+---+---+---+---+---+
```

The sender specific IV (SSIV) is a packet counter.

Group Size

The group size is the length of the SID space allocation for KSSIDs as well as GMSIDs that are reserved to a KS for distribution to GMs. Available group sizes are small (8, 12, or 16 bits), medium (24 bits, which is the default), and large (32 bits). Medium is sufficient for nearly all networks.

You should use a large group size only if you must strictly adhere to the requirement in section A.5, “Key/IV Pair Uniqueness Requirements from SP 800-38D” of the publication Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program in which GET VPN used in conjunction with Suite B must have at least 2^{32} unique possible “module names” (SIDs).This publication is issued and maintained by the NIST and the Communications Security Establishment Canada (CSEC).

For example, in a large group size with one KS, the SID is 32 bits, there are 512 KSSID values (in the range of 0 to 511), and each has 8,388,607 GMSIDs to distribute to registering GMs. With a large group size, use the following KSSID assignment guidelines to configure KSSID ranges:

Table 298: Recommended KSSID Ranges for Group Size Large

KS	1 KS (no cooperative KSs)	2 cooperative KSs	3 cooperative KSs	4 cooperative KSs
KS1	0 - 511	0 - 255	0 - 127	0 - 63

KS	1 KS (no cooperative KSs)	2 cooperative KSs	3 cooperative KSs	4 cooperative KSs
KS2	—	256 - 511	128 - 255	64 - 127
KS3	—	—	256 - 383	128 - 191
KS4	—	—	384 - 511	192 - 255
KS5	—	—	—	256 - 319
KS6	—	—	—	320 - 383
KS7	—	—	—	384 - 447
KS8	—	—	—	448 - 511

If you plan to expand the cooperative KS network to include more KSs, while you are initially configuring the original KS or KSs, use the column in the above table with the *anticipated* number of KSs in the network so that you can add the new KS or KSs later.

You should use a small (8-, 12-, or 16-bit) group size only in well-understood cases where strict interoperability with SID lengths of 8, 12, and 16 bits is required according to RFC 6054, Using Counter Modes with Encapsulating Security Payload (ESP) and Authentication Header (AH) to Protect Group Traffic . If such interoperability is needed, you must be careful when designing the network, because the number of SIDs per group is severely limited (and therefore, the number of KSs and GMs in a group is severely limited). Following are the limitations for a small group size:

Table 299: Limitations for Group Size Small

SID length	KSSIDs (total KSs)	GMSIDs per KSSID	GMSIDs (total GMs)	Possible number of GM registrations for one KS (after assigning KSSIDs to all KSs evenly)			
—	—	—	—	**1 KS**	**2 KSs**	**4 KSs**	**8 KSs**
8 bits	2	128	255	320	96	—	—
12 bits	4	1,024	4,095	3,840	1,792	768	—
16 bits	16	4,096	65,535	64,512	31,744	15,360	7,168

KSSID Assignment with Cooperative Key Servers

You should plan ahead to assign a certain number of initial GDOI KS identifiers (KSSIDs) to each KS based on the configured group size, number of KSs, number of GMs, number of GMs per KS, and any future expansion of KSs or GMs (or both).

When there are multiple cooperative KSs in a GDOI group, each KS must have a unique set of KSSID values to ensure that a registering GM never receives the same SID as another registering GM in the group. Therefore, you should plan how you will assign KSSIDs across cooperative KSs in advance, while considering the number of cooperative KSs and if cooperative KSs will be added later. If none will be added, you can assign all available KSSIDs across all KSs. If cooperative KSs will be added, you should reserve some KSSIDs to assign to those KSs when you add them to the network.

You can reassign KSSIDs; however, if KSSIDs that are already used by a KS to distribute GMSIDs are removed from the KS, the group will reinitialize (meaning that all GMs will be forced to re-register, and TEK IPsec SAs will be rekeyed to reset the used KSSIDs) without traffic loss. To avoid this group reinitialization, use the guidelines in the following table (which uses the default group size of medium):

Table 300: Recommended KSSID Assignment Ranges for Cooperative KSs (Group Size Medium)

	1 KS (no cooperative KSs)	2 cooperative KSs	3 cooperative KSs	4 cooperative KSs
KS1	0 - 127	0 - 63	0 - 31	0 - 15
KS2	—	64 - 127	32 - 63	16 - 31
KS3	—	—	64 - 95	32 - 47
KS4	—	—	96 - 127	48 - 64
KS5	—	—	—	65 - 80
KS6	—	—	—	81 - 95
KS7	—	—	—	96 - 112
KS8	—	—	—	113 - 127

If you plan to expand the cooperative KS network to include more KSs, when initially configuring the original KS (or KSs), use the column in the above table with the planned number of KSs in the *expanded* network so that the new KS or KSs can be added later.

Following are additional guidelines for assigning KSSIDs to KSs:

- Configure only contiguous blocks of KSSIDs across KSs (for example, KS1 = 0-9 + 40-49, KS2 = 10-19 + 50-59, KS3 = 20-29, KS4 = 30-39, and so on).
- Any one KS should have enough KSSID space to receive all GM registrations from the group (in case the other KSs fail all of their GM registrations).
- To avoid reinitialization of the group, only add new KSSID values or ranges; do not remove them unless necessary.
- During a network split (a connectivity loss among cooperative KSs), do not change the KSSID assignment; this prevents overlapping KSSIDs, which would cause reinitialization on a merge (when connectivity has been restored among cooperative KSs).
- If the group begins in an *n*-way split (meaning that secondary KSs are planned but not yet configured), configure all of the KSSIDs as if the group was not split.

The number of KSSIDs available depends on the group size configuration as in the following table:

Table 301: Ranges of Available KSSIDs Based on Group Size

Configured Group Size	Number of Available KSSIDs
Small (8-bit)	0 to 1

Configured Group Size	Number of Available KSSIDs
Small (12-bit)	0 to 3
Small (16-bit)	0 to 15
Medium	0 to 127
Large	0 to 511

Group Reinitialization

Group reinitialization is the process of retiring KSSIDs. Group reinitialization occurs across all KSs (primary and secondary). Any KS can trigger a group reinitialization, and it occurs whenever

- You change the TEK policy from non-GCM to GCM.
- You change the group size.
- You remove a previously used KSSID.
- A KS in the group runs out of both KSSIDs and GMSIDs.
- A KSSID overlap that was detected by a cooperative KS is resolved.

During reinitialization, all KSs move their used KSSIDs to old (used) KSSIDs (and they are thus retired). Then, reinitialization creates a new KEK and new TEKs, lowers the existing TEK lifetime, and deletes the existing TEKs to cause all GMs to re-register (within the window determined by the **clear crypto gdoi ks members** command). This window is five percent of the remaining lifetime, between 90 seconds and one hour. When the lifetime of the existing TEKs has expired, each KS resets its old (used) KSSIDs, then all KSSIDs are available for use once again.

Reinitialization does not cause traffic disruption on the GMs. All GMs receive new GMSIDs with new TEKs when re-registering.

Cisco GET VPN System Logging Messages for Suite B

The tables below explain the GET VPN system logging (also called syslog) messages that are related to Suite B.

Table 302: KS and Cooperative KS Messages

Message	Explanation
%GDOI-5-KS_REINIT_GROUP: *reason* for group *group-name* and will re-initialize the group.	The KS will reinitialize the group. The possible *reason* strings are as follows: • KS configured Suite-B transform requiring SIDs • KS configured Suite-B transform requiring SIDs during scheduled rekey • KS is running out of SIDs • KS changed Group Size • KS removed used KSSIDs • KS issued 'clear crypto gdoi ks members' • KS issued re-init test cmd • KSSID overlap was resolved • Pri KS peer changed used Group Size • Pri KS peer sent re-init request • Sec KS peer sent re-init request
%GDOI-5-KS_REINIT_FINISH: Re-initialization of group *group-name* completed.	Reinitialization for the group is complete. It is useful to know when a reinitialization has completed, because some operations are blocked during a reinitialization (such as when the group size is changed and used KSSIDs are removed). A reinitialization does not finish until the old (used) TEK is cleared, which might not occur until a reinitialization is checked again (for example while a **show** command is executing, while a group size or KSSIDs are being configured, or when a cooperative KS is being updated) or until the next GM registers.
%GDOI-3-KS_NO_SID_AVAILABLE: GMs for group *group-name* need SIDs but this KS has no KS SIDs configured or no more SIDs available.	(When using GCM and after a GM begins registration) GMs for the group need SIDs, but either the KS has no KSSIDs configured or has no more SIDs available.

Message	Explanation
%GDOI-3-COOP_KS_KSSID_OVERLAP: Overlapping KS Sender Identifier(s) (KSSID) {*KSSID*|*KSSID-Range*} with COOP-KS peer *ip-address* in group *group-name* blocking GM registration (MISCONFIG).	A KSSID or KSSID range that overlaps with a cooperative KS peer in another group is blocking GM registration. An overlapping KSSID configuration is blocked on cooperative KSs by the CLI, but it might occur in a GET VPN network split scenario (in which one or more cooperative KSs were temporarily unavailable but have come back online) or with saved configurations.
%GDOI-5-COOP_KS_KSSID_OVERLAP_RESOLVED: Resolved overlapping KS Sender Identifier(s) (KSSID) with COOP-KS peer allowing GM registrations once again.	A KSSID that overlaps with a cooperative KS peer was resolved (which allows GM registrations to resume).

Table 303: GM Messages

Message	Explanation
%GDOI-5-GM_IV_EXHAUSTED: GM for group *group-name* exhausted its IV space for interface *interface-name* and will re-register.	The GM for the group exhausted its IV space (meaning its set of unique IVs) for a particular SA and will re-register.
%GDOI-5-GM_REJECTING_SA_PAYLOAD: Registration: Policy in SA payload sent by KS *ip-address* rejected by GM in the group *group-name* reason: client rekey hash algorithm (*kek-policy*) is unacceptable by this GM.	The client rekey hash algorithm (the specified KEK policy) was not accepted by a GM in the specified group. At registration, the GM rejected the KEK policy.
%GDOI-5-GM_REJECTING_SA_PAYLOAD: Registration: Policy in SA payload sent by KS *ip-address* rejected by GM in the group *group-name* reason : client rekey transform-sets (*tek-policy*) for data-protection are unacceptable by this GM.	The client rekey transform sets (the specified TEK policy) for data protection was not accepted by the GM. At registration, the GM rejected the TEK policy.
%GDOI-5-GM_REKEY_TRANSFORMSET_CHECK_FAIL: The transform set (*transform-set*) for data protection in group *group-name* is unacceptable by this client.	The transform set for data protection in the group was not accepted by the client. The GM received a rekey and rejected the TEK policy.
%GDOI-3-KS_REKEY_AUTH_KEY_LENGTH_INSUFFICIENT: Rejected rekey sig-hash algorithm change: using sig-hash algorithm HMAC_AUTH_SHA*bits* requires an authentication key length of at least *number-of-bits* bits (*number-of-blocks* blocks in bytes) - current RSA key "360-bit" is only 45 blocks in bytes.	Configuration of the rekey signature hash algorithm was rejected, because the RSA key did not have a long enough modulus. HMAC-SHA-384 requires a modulus of at least 465 bits (59 blocks in bytes), and HMAC-SHA-512 requires a modulus of 593 bits (75 blocks in bytes).

How to Configure GET VPN Support with Suite B

Each feature in the GET VPN Support with Suite B feature set is independently configurable. But to be compliant with the Suite B standard, you must configure certain combinations of these features. For more information about these combinations, see RFC 4869, Suite B Cryptographic Suites for IPsec .

Ensuring that GMs Are Running Software Versions That Support Suite B

Because GET VPN is a technology that is based on groups, all devices in the same group (including the primary KS, cooperative KSs, and GMs) must support the Suite B feature before you can enable the feature. If you want to enable the feature for a group, you must ensure that all devices in the group are running compatible versions of the GET VPN software.

To ensure that all devices in the GET VPN network support Suite B, perform the following steps on the KS (or primary KS).

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature suite-b**
3. **show crypto gdoi feature suite-b | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature suite-b** **Example:** `Device# show crypto gdoi feature suite-b`	Displays the version of the GET VPN software running on each KS and GM in the network and displays whether that device supports Suite B.
Step 3	**show crypto gdoi feature suite-b | include No** **Example:** `Device# show crypto gdoi feature suite-b | include No`	(Optional) Finds only those devices that do not support Suite B.

Configuring a Key Server for GET VPN Suite B

Configuring the Signature Hash Algorithm for the KEK

Perform this task to configure the signature hash algorithm for the KEK.

Before you begin

This task has the following prerequisites:

- Make sure that rekey authentication that is using an RSA key pair associated with the device is enabled. To do so, use the **rekey authentication** command with the **mypubkey rsa** *key-name* keywords and argument.
- Make sure that the RSA key pair has a modulus of sufficient length. HMAC-SHA-384 requires a modulus of at least 465 bits (59 blocks in bytes), and HMAC-SHA-512 requires a modulus of 593 bits (75 blocks in bytes). If the rekey signature hash algorithm is changed to SHA-384 or SHA-512 with a key pair of insufficient modulus length, a configuration rejection message appears on the console, and system logging messages are generated. Similarly, if the rekey signature hash algorithm is already SHA-384 or SHA-512 and the key pair is modified to one of insufficient modulus length, a similar message appears on the console, and the same system logging messages are generated.
- To use SHA-2/HMAC-SHA-2 for authentication of the *acknowledgment* from GMs to KSs after receiving a rekey message, you must enable unicast distribution of rekey messages to GMs. To do so, use the **rekey transport unicast** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **rekey sig-hash algorithm** *algorithm*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:**	Designates a device as a GDOI KS and enters GDOI local server configuration mode.

	Command or Action	Purpose
	`Device(config-gdoi-group)# server local`	
Step 5	**rekey sig-hash algorithm** *algorithm* **Example:** `Device(gdoi-local-server)# rekey sig-hash algorithm sha512`	Configures the signature hash algorithm for the KEK. For Suite B, you must specify **sha256**, **sha384**, or **sha512**.
Step 6	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring the Group Size

This task is optional. For nearly all deployments, the default group size (sender identifier length) of medium is recommended. Perform this task to configure the group size for Suite B.

When you change the group size in a group with cooperative KSs after Suite B (meaning ESP-GCM or ESP-GMAC) is configured and after the Suite B policy has been generated, you must change the group size on all secondary KSs before changing it on the primary KS.

Changing the group size causes the group to reinitialize (so that the new SID length can be used). Conflicting group size configurations across KSs will block GM registration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **group size** {**small** {**8** | **12** | **16**} | **medium** | **large**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 5	**group size** {**small** {**8** | **12** | **16**} | **medium** | **large**} **Example:** `Device(gdoi-local-server)# group size small 16`	Configures the group size.
Step 6	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring Key Server Identifiers

Suite B requires the assignment of unique GMSIDs to each GM, which means that a GM cannot reuse a previously used SID (either from itself or another GM) for the same key. Therefore, although GET VPN is designed to disallow overlapping SID values, you should correctly configure KSSID values among KSs so that each KS has a unique set. (KSSID overlap among KSs will cause a reinitialization.)

You must configure at least one unique KSSID to allot a pool of SIDs to the KS. You do so on the KS before configuring GCM or GMAC as the TEK IPsec policy.

Perform this task to assign a KSSID or a range of KSSIDs to a KS. Each KS must be assigned at least one KSSID when using GCM or GMAC. You can configure a single KSSID, a range of KSSIDs, or both. For the default group size of medium, there are 128 possible KSSID values in the range from 0 to 127.

KSSID values are not assigned to (and usable by) the KS until you exit GDOI local server ID configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **identifier**
6. **range** *lowest-kssid* **-** *highest-kssid*
7. **value** *kssid*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 5	**identifier** **Example:** `Device(gdoi-local-server)# identifier`	Enters GDOI local server ID configuration mode.
Step 6	**range** *lowest-kssid - highest-kssid* **Example:** `Device(gdoi-local-server-id)# range 10 - 20`	Assigns a range of KSSIDs. • This range must be unique in the entire group.
Step 7	**value** *kssid* **Example:** `Device(gdoi-local-server-id)# value 0`	Assigns a KSSID. • This KSSID must be unique in the entire group. • The **value 0** command allots the pool of SIDs to the KS that begin with KSSID value 0 (meaning that it is allotted the pool of SID values beginning with 0x0 and ending with 0x1FFFF).
Step 8	**end** **Example:** `Device(gdoi-local-server-id)# end`	Exits GDOI local server ID configuration mode and returns to privileged EXEC mode.

If you try to configure one or more KSSIDs on a KS that are already assigned to another KS (and the cooperative KS network is not split), the configuration is denied, and the following message appears when you exit GDOI local server ID configuration mode:

```
% Key Server SID Configuration Denied:
%   The following Key Server SIDs being added overlap:
%   2, 200-250  (COOP-KS Peer: 10.0.9.1)
```

If the cooperative KS network *is* split, you should not configure overlapping KSSIDs. If overlapping KSSIDs are detected on a network merge, GM registration is blocked until the overlap is resolved. The following system logging message appears on both KSs:

```
%GDOI-3-COOP_KSSID_OVERLAP: Overlapping KS Sender Identifier(s) (KSSID) {2, 200-250} with
COOP-KS peer 10.0.9.1 in group diffint blocking GM registration (MISCONFIG)
```

When a KS unconfigures the overlapping KSSIDs, the group reinitializes (meaning that all GMs are forced to re-register, and TEK IPsec SAs are rekeyed to reset the used KSSIDs) without traffic loss. The following system logging messages appear on the KS:

```
%SYS-5-CONFIG_I: Configured from console by console
%GDOI-5-COOP_KSSID_OVERLAP_RESOLVED: Resolved overlapping KS Sender Identifier(s) (KSSID)
with COOP-KS peer allowing GM registrations once again
%GDOI-5-KS_REINIT_GROUP: KSSID overlap was resolved for group diffint and will re-initialize
 the group.
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group diffint from address 10.0.8.1
 with seq # 11
%GDOI-4-GM_DELETE: GM 10.0.3.1 deleted from group diffint.
%GDOI-4-GM_DELETE: GM 10.65.9.2 deleted from group diffint.
```

The %GDOI-5-KS_SEND_UNICAST_REKEY system logging message appears only if this is the primary KS. The peer KS that had overlapping KSSIDs also displays the %GDOI-5-COOP_KSSID_OVERLAP_RESOLVED system logging message.

Configuring the IPsec SA for Suite B

To configure the IPsec SA for Suite B, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**]
4. **crypto ipsec profile** *ipsec-profile-name*
5. **set transform-set** *transform-set-name*
6. **exit**
7. **crypto gdoi group** [**ipv6**] *group-name*
8. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
9. **server local**
10. **sa ipsec** *sequence-number*
11. **profile** *ipsec-profile-name*

12. **match address** {**ipv4** | **ipv6**} {*access-list-number* | *access-list-name*}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**] **Example:** `Device(config)# crypto ipsec transform-set g1 esp-gcm 192`	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode. • For Suite B, you must specify a transform set using ESP-GCM or ESP-GMAC. (You can define multiple transform sets by entering the command again on separate command lines.) • You can optionally specify a key size of 128, 192, or 256. The default key size is 128.
Step 4	**crypto ipsec profile** *ipsec-profile-name* **Example:** `Device(config)# crypto ipsec profile profile1`	Defines the IPsec profile (the parameters to be used for IPsec encryption between two IPsec routers) and enters IPsec profile configuration mode.
Step 5	**set transform-set** *transform-set-name* **Example:** `Device(ipsec-profile)# set transform-set transformset1`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile configuration mode.
Step 7	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.

	Command or Action	Purpose

Step 8	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.
		• The **identity number** *number* command applies to IPv4 and IPv6 configurations.
		• The **identity address ipv4** *address* command applies only to IPv4 configurations.

Step 9	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 10	**sa ipsec** *sequence-number* **Example:** `Device(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 11	**profile** *ipsec-profile-name* **Example:** `Device(gdoi-sa-ipsec)# profile gdoi-p`	Defines the IPsec SA policy for a GDOI group.
Step 12	**match address** {**ipv4** | **ipv6**} {*access-list-number* | *access-list-name*} **Example:** `Device(gdoi-sa-ipsec)# match address ipv4 102`	Selects an IP extended access list (ACL) for a GDOI registration. • You must use the **ipv4** keyword for IPv4 groups and the **ipv6** keyword for IPv6 groups. • You must use a named (not numbered) access list for IPv6 configurations. **Note** Make sure that you select an ACL that has identical entries in the identical order among all the cooperative KSs for the group. If not, GMs that register to separate KSs cannot encrypt and decrypt correctly after downloading the policy.

	Command or Action	Purpose

		Note If you attempt to assign an IPv6 group with IPv4 policies, an error message appears indicating that the access list name is invalid, or the list already exists but is the wrong type:
		`Access-list type conflicts with prior definition`
		`% ERROR: access-list-name is either an invalid name or the list already exists but is the wrong type.`

Step 13	**end** **Example:** `Device(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

Configuring a Group Member for GET VPN Suite B

Configuring Acceptable Ciphers or Hash Algorithms for KEK for Suite B

To configure the Suite B ciphers and hash algorithms for KEK to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server address ipv4** *address*
6. **client rekey encryption** *cipher* [... [*cipher*]]
7. **client rekey hash** *hash*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server address ipv4** *address* **Example:** `Device(config-gdoi-group)# server address ipv4 10.0.5.2`	Specifies the address of the server that a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.
Step 6	**client rekey encryption** *cipher* [... [*cipher*]] **Example:** `Device(config-gdoi-group)# client rekey encryption 3des-cbc aes 192 aes 256`	Sets the client acceptable rekey ciphers for the KEK.
Step 7	**client rekey hash** *hash* **Example:** `Device(config-gdoi-group)# client rekey hash sha384`	Sets the client acceptable hash algorithm for KEK. • For Suite B, you must specify either **sha256**, **sha384**, or **sha512**.
Step 8	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring Acceptable Transform Sets for TEKs for Suite B

To configure the transform sets used by TEKs for data encryption or authentication to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**]
4. **exit**
5. **crypto gdoi group** [**ipv6**] *group-name*
6. **client transform-sets** *transform-set-name1* [... [*transform-set-name6*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**] **Example:** `Device(config)# crypto ipsec transform-set g1 esp-gcm 192`	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode. • For Suite B, you must specify a transform set using ESP-GCM or ESP-GMAC. • You can define multiple transform sets by entering the command again on separate command lines. • You can optionally specify a key size of 128, 192, or 256. The default key size is 128.
Step 4	**exit** **Example:** `Device(cfg-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 5	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.

	Command or Action	Purpose
Step 6	**client transform-sets** *transform-set-name1* [... [*transform-set-name6*]] **Example:** `Device(config-gdoi-group)# client transform-sets g1`	Specifies the acceptable transform-set tags used by TEKs for data encryption and authentication. • You can specify up to six transform-set tags.
Step 7	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Verifying and Troubleshooting GET VPN Support with Suite B

Verifying and Troubleshooting GET VPN Support with Suite B on a Key Server

To view the configuration that is running on a KS, use the **show running-config** command.

SUMMARY STEPS

1. **show crypto gdoi ks identifier** [**detail**]
2. **show crypto gdoi ks coop identifier** [**detail**]
3. **show crypto gdoi feature suite-b**
4. **show crypto gdoi ks policy**

DETAILED STEPS

Step 1 **show crypto gdoi ks identifier** [**detail**]

Example:

```
Device# show crypto gdoi ks identifier detail

KS Sender ID (KSSID) Information for Group diffint:

    Transform Mode           : Counter (Suite B)
    reinitializing          : No
    SID Length (Group Size)  : 24 bits (medium)
    Current KSSID In-Use     : 0
    Last GMSID Used          : 1

    KSSID (or SIDS)Assigned        : 0-15
    KSSID (or SIDS)Used            : 0
    KSSID (or SIDS) Used  (Old)      : none
    Available KSSID (or SIDS): 1-15

    REMAINING SIDs:
      KSSID to reinitialize at         : 15
      GMSID to reinitialize at         : 6291456
      # of SIDs Remaining for Cur KSSID : 8388606
```

```
    # of SIDs Remaining until Re-init : 132120575
```

This command displays the status of SID management for Suite B. The Transform Mode field can be either Non-Counter (Non-Suite B) or Counter (Suite B) to check if SID management and a Suite B policy is currently used in the group. If the group is currently reinitializing (meaning that all GMs will be forced to re-register, and TEK IPsec SAs will be rekeyed to reset the used KSSIDs), then the reinitializing field displays Yes. The SID Length (Group Size) field determines the group size currently used in the group, which defaults to 24 bits (medium).

The Current KSSID In-Use and Last GMSID Used fields correspond to the SID (or SIDS) to be distributed to the next registering GM. The KSSID (or SIDS) Assigned field corresponds to the locally configured KSSIDs that have been synced with cooperative KSs, and the Available KSSID (or SIDS) field corresponds to those KSSIDs that have not been used yet since the last reinitialization. Each time a new KSSID is used, it is added to the KSSID (or SIDS) Used field, and during a reinitialization, those used KSSIDs are transferred to the KSSID (or SIDS) Used (Old) field. At the end of a reinitialization period, the old used KSSIDs are cleared and put in the Available KSSIDs pool again.

Note When the value in the # of SIDs Remaining until Re-init field approaches 0, a reinitialization will occur soon if GMs are continuing to re-register. Although a reinitialization should not cause traffic disruption or network problems, it will cause all GMs to re-register.

Step 2 **show crypto gdoi ks coop identifier** [**detail**]

Example:

```
Device# show crypto gdoi ks coop identifier detail

COOP-KS Sender ID (SID) Information for Group diffint:

    Local KS Role: Primary   , Local KS Status: Alive
        Local Address                : 10.0.8.1
        Next SID Client Operation    : NOTIFY
        reinitializing              : No
        KSSID Overlap                : No
        SID Length (Group Size) Cfg  : 24 bits (medium)
        SID Length (Group Size) Used : 24 bits (medium)
        Current KSSID In-Use         : 0
        KSSID (or SIDS)Assigned            : 0-15
        KSSID (or SIDS)Used                : 0
        Old KSSID (or SIDS)Used            : none

    Peer KS Role:  Secondary , Peer KS Status:  Alive
        Peer Address                 : 10.0.9.1
        Next SID Client Operation    : NOTIFY
        reinitializing              : No
        KSSID Overlap                : No
        SID Length (Group Size) Cfg  : 24 bits (medium)
        SID Length (Group Size) Used : 24 bits (medium)
        Current KSSID In-Use         : 16
        KSSID (or SIDS)Assigned            : 16-31
        KSSID (or SIDS)Used                : 16
        Old KSSID (or SIDS)Used            : none
```

This command displays the status of SID information that is synchronized across cooperative KSs.

When the KSSID Overlap field displays Yes, GM registration is blocked until the overlap of KSSIDs (which could have happened during a network split) is resolved. You must unconfigure the overlapping KSSIDs from one cooperative KS or the other before GM registration can resume. When the overlapping KSSIDs are resolved, a reinitialization occurs.

When you change the group size (not recommended for most deployments), all secondary KSs must first configure the new group size. Then on the primary KS, the SID Length (Group Size) Cfg field displays the new group size on all cooperative KS peers. Only when the primary KS configures the new group size will all KSs start to use the new group size and update the SID Length (Group Size) Used field to display the new group size.

Step 3 **show crypto gdoi feature suite-b**

Example:

```
Device# show crypto gdoi feature suite-b

Group Name: diffint
    Key Server ID       Version   Feature Supported
        10.0.8.1         1.0.4          Yes
        10.0.9.1         1.0.4          Yes

    Group Member ID     Version   Feature Supported
        10.0.3.1         1.0.4          Yes
        10.0.4.1         1.0.4          Yes
```

This command displays whether KSs and GMs can use the Suite B feature set (meaning AES-GCM, AES-GMAC, SHA-2, and HMAC-SHA2). The Version field must display 1.0.4 or higher, and the Feature Supported field must display Yes for all KSs in the cooperative KS group and for the registered GMs.

Step 4 **show crypto gdoi ks policy**

Example:

```
Device# show crypto gdoi ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):

  # of teks : 4  Seq num : 0
  KEK POLICY (transport type : Unicast)
    spi : 0x80474E999FE8F60364B7F51809E28C84
    management alg     : disabled    encrypt alg       : 3DES
    crypto iv length   : 8           key size          : 24
    orig life(sec): 86400       remaining life(sec): 85586
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys

  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x9C666FA7
    access-list        : gcm-acl
    Selector           : permit ip host 10.0.1.1 host 239.0.1.1
    transform          : esp-gcm
    alg key size       : 20            sig key size          : 0
    orig life(sec)     : 900           remaining life(sec)   : 87
    tek life(sec)      : 900           elapsed time(sec)     : 813
    override life (sec): 0             antireplay window size: 64


  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x54E8D5D3
    access-list        : gcm-acl
    Selector           : permit ip host 10.0.100.2 host 238.0.1.1
    transform          : esp-gcm
    alg key size       : 20            sig key size          : 0
```

```
  orig life(sec)     : 900           remaining life(sec)   : 87
  tek life(sec)      : 900           elapsed time(sec)     : 813
  override life (sec): 0             antireplay window size: 64


TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi                : 0xC8B4DE6D
  access-list        : gcm-acl
  Selector           : permit ip host 10.0.1.1 host 10.0.100.2
  transform          : esp-gcm
  alg key size       : 20            sig key size          : 0
  orig life(sec)     : 900           remaining life(sec)   : 87
  tek life(sec)      : 900           elapsed time(sec)     : 813
  override life (sec): 0             antireplay window size: 64


TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi                : 0x1C908AF3
  access-list        : gcm-acl
  Selector           : permit ip host 10.0.100.2 host 10.0.1.1
  transform          : esp-gcm
  alg key size       : 20            sig key size          : 0
  orig life(sec)     : 900           remaining life(sec)   : 87
  tek life(sec)      : 900           elapsed time(sec)     : 813
```

This command displays whether a TEK and IPsec SA were generated per ACE (displayed in the Selector field) from the ACL in the access-list field for the ESP-GCM or ESP-GMAC TEK policy. This command also displays whether the KEK policy is using SHA-2/HMAC-SHA-2 as the signature hash algorithm.

Verifying and Troubleshooting GET VPN Support with Suite B on a GM

To view the configuration that is running on a GM, use the **show running-config** command.

SUMMARY STEPS

1. **show crypto gdoi gm identifier** [**detail**]
2. **show crypto gdoi feature suite-b**
3. **show crypto gdoi**

DETAILED STEPS

Step 1 **show crypto gdoi gm identifier** [**detail**]

Example:

```
Device# show crypto gdoi gm identifier detail

GM Sender ID (SID) Information for Group diffint:

  Group Member: 10.65.9.2        vrf: None
    Transform Mode                  : Counter (Suite B)
    # of SIDs Last Requested        : 3
```

```
    CURRENT SIDs:
      Shared Across Interfaces?     : Yes
      SID Length (Group Size)       : 24 bits (medium)
      # of SIDs Downloaded          : 3
      First SID Downloaded          : 0x08000007
      Last SID Downloaded           : 0x08000009

      CM Interface   B/W (Kbps) MTU (B)  # Req # Rx  Installed SID Range
      ============== ========== ======== ===== ===== ======================
      Et2/0          10000      1500     1     3     0x08000007 - 0x08000009
      Et3/0          10000      1500     1     3     0x08000007 - 0x08000009
      Et4/0          10000      1500     1     3     0x08000007 - 0x08000009

    NEXT SID REQUEST:
      TEK Lifetime                  : 900 sec
      SID Length (Group Size)       : 32 bits (LARGE)
```

This command displays the status of received and installed SIDs on a GM when it is using GCM-AES or GMAC-AES as the TEK IPsec SA policy. The Transform Mode field can display Non-Counter (Non-Suite B) or Counter (Suite B) to check whether SIDs are being downloaded and installed and whether a Suite B policy is used in the group. The # of SIDs Last Requested field mainly depends on the number of interfaces to which the crypto map is applied for this registered GM (meaning using the local-address or client registration interface). The SIDs are Shared Across Interfaces field when using local-address and each CM Interface's Installed SID Range field will be the same. You use this command mainly to verify that each CM interface has SIDs installed.

Step 2 **show crypto gdoi feature suite-b**

Example:

```
Device# show crypto gdoi feature Suite B

       Version    Feature Supported
        1.0.4           Yes
```

This command displays whether this GM can use the Suite B feature set (meaning GCM-AES, GMAC-AES, SHA-2, and HMAC-SHA-2). The Version field must display 1.0.4 or higher, and the Feature Supported field must display Yes.

Step 3 **show crypto gdoi**

Example:

```
Device# show crypto gdoi

GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 1234
    Crypto Path              : ipv4
    Key Management Path      : ipv4
    Rekeys received          : 0
    IPSec SA Direction       : Both

     Group Server list       : 10.0.8.1

    Group member             : 10.0.3.1         vrf: None
       Version               : 1.0.4
       Registration status   : Registered
       Registered with       : 10.0.8.1
```

```
.
.
.
ACL Downloaded From KS 10.0.8.1:
   access-list   permit ip host 10.0.1.1 host 239.0.1.1
   access-list   permit ip host 10.0.100.2 host 238.0.1.1
   access-list   permit ip host 10.0.1.1 host 10.0.100.2
   access-list   permit ip host 10.0.100.2 host 10.0.1.1

KEK POLICY:
    Rekey Transport Type     : Unicast
    Lifetime (secs)          : 85740
    Encrypt Algorithm        : 3DES
    Key Size                 : 192
    Sig Hash Algorithm       : HMAC_AUTH_SHA256
    Sig Key Length (bits)    : 1024

TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet3/0:
    IPsec SA:
        spi: 0x318846DE(831014622)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xF367AEA0(4083658400)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xE583A3F5(3850609653)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xE9AC04C(245022796)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64
```

The presence of multiple IPsec SAs shows that GCM or GMAC is configured (note that each IPsec SA has a unique SPI for each ACE that was downloaded). For each ACE listed in the TEK POLICY for the current KS-Policy ACEs Downloaded section, this command displays whether a TEK policy and IPsec SA were downloaded (and installed) from the ACLs that are listed in the ACL Downloaded From KS section. This command also displays whether the KEK policy is using SHA-2/HMAC-SHA-2 for the signature hash algorithm (for example, HMAC_AUTH_SHA256).

Configuration Examples for GET VPN Support with Suite B

Example: Ensuring that GMs Are Running Software Versions That Support Suite B

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support Suite B cryptography:

```
Device# show crypto gdoi feature suite-b

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.4          Yes
        10.0.6.2         1.0.4          Yes
        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No

    Group Member ID     Version   Feature Supported
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
        10.0.3.1         1.0.4          Yes
        10.0.3.2         1.0.4          Yes
```

You can also enter the above command on a GM (which will display the information for the GM but not for the KS or other GMs).

The following example shows how to enter the command on the KS (or primary KS) find only those devices in the GET VPN network that do *not* support Suite B:

```
Device# show crypto gdoi feature suite-b | include No

        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
```

Example: Configuring a Key Server for GET VPN Suite B

Configuring the Signature Hash Algorithm for the KEK

The following example shows how to configure the signature hash algorithm for the KEK:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey sig-hash algorithm sha512
Device(gdoi-local-server)# end
```

Configuring the Group Size for Suite B

Configuring the group size for Suite B is optional, because the default group size of medium is sufficient for most deployments. The following example shows how to configure the group size for Suite B:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# group size small 16
Device(gdoi-local-server)# end
```

Configuring Key Server Identifiers

The following example shows how to assign a KSSID as well as a range of KSSIDs to a KS:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# identifier
Device(gdoi-local-server-id)# range 10 - 20
Device(gdoi-local-server-id)# value 0
Device(gdoi-local-server-id)# end
```

Configuring the IPsec SA for Suite B

The following example shows how to configure the IPsec SA for Suite B. This example uses an identity number instead of an identity address:

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec transform-set g1 esp-gcm 192
Device(config)# crypto ipsec profile profile1
Device(ipsec-profile)# set transform-set transformset1
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group gdoigroupname
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# profile gdoi-p
Device(gdoi-sa-ipsec)# match address ipv4 102
Device(gdoi-sa-ipsec)# end
```

Example: Configuring a Group Member for GET VPN Suite B

Configuring Ciphers or Hash Algorithms for the KEK for Suite B

The following example shows how to configure the Suite B ciphers and hash algorithms for the KEK to be allowed by the GM. This example uses an identity address (compatible only with IPv4 data plane configurations). You could instead use an identity number (which would be compatible with IPv4 and IPv6 data plane configurations).

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group gdoigroupone
Device(config-gdoi-group)# identity address ipv4 10.2.2.2
Device(config-gdoi-group)# server address ipv4 10.0.5.2
Device(config-gdoi-group)# client rekey encryption 3des-cbc aes 192 aes 256
Device(config-gdoi-group)# client rekey hash sha384
Device(config-gdoi-group)# end
```

Configuring Acceptable Transform Sets for TEKs for Suite B

The following example shows how to configure the acceptable transform sets used by TEKs for data encryption or authentication.

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec transform-set g1 esp-gcm 192
Device(cfg-crypto-trans)# exit
Device(config)# crypto gdoi group gdoigroupone
Device(config-gdoi-group)# client transform-sets g1
Device(config-gdoi-group)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
IKE and IKE policy configuration tasks IPsec transform configuration tasks	"Configuring Internet Key Exchange for IPsec VPNs" module in the Internet Key Exchange for IPsec VPNs Configuration Guide, Cisco IOS Release 15.2M&T
Basic deployment guidelines for enabling GET VPN in an enterprise network	Cisco IOS GET VPN Solutions Deployment Guide

Standards and RFCs

Standard/RFC	Title
Federal Information Processing Standard (FIPS) Publication 140-2	Security Requirements for Cryptographic Modules
RFC 2401	Security Architecture for the Internet Protocol
RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
RFC 4543	The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
RFC 4869	Suite B Cryptographic Suites for IPsec
RFC 6054	Using Counter Modes with Encapsulating Security Payload (ESP) and Authentication Header (AH) to Protect Group Traffic
RFC 6407	The Group Domain of Interpretation

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GET VPN Support with Suite B

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 304: Feature Information for GET VPN Support with Suite B

Feature Name	Releases	Feature Information
GET VPN Support with Suite B		The GET VPN Support with Suite B feature adds support of the Suite B set of ciphers to Cisco Group Encrypted Transport (GET) VPN. Suite B is a set of cryptographic algorithms that includes Galois Counter Mode Advanced Encryption Standard (GCM-AES) as well as algorithms for hashing, digital signatures, and key exchange. Suite B for IP security (IPsec) VPNs is a standard whose usage is defined in RFC 4869. Suite B provides a comprehensive security enhancement for Cisco IPsec VPNs, and it allows additional security for large-scale deployments. Suite B is the recommended solution for organizations requiring advanced encryption security for the wide-area network (WAN) between remote sites. The following commands were introduced or modified: **client rekey hash, crypto key export ec, crypto key generate ec keysize, crypto key import ec, group size, identifier, rekey sig-hash algorithm, show crypto gdoi.**

CHAPTER 226

GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

The Cisco TrustSec (CTS) architecture secures networks by establishing domains of trusted network devices. Once a network device authenticates with the network, the communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and replay protection mechanisms.

CTS uses the user and device identification information acquired during the authentication phase to classify packets as they enter the network. CTS maintains classification of each packet or frame by tagging it with a security group tag (SGT) on ingress to the network so that it can be identified for applying security and other policy criteria along the data path. The tags allow network intermediaries such as switches and firewalls to enforce access control policy based on the classification.

The GET VPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GET VPN inline tagging to carry the SGT information across the private WAN.

Prerequisites for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.5 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support it.

This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support IPsec inline tagging for Cisco TrustSec. For more information, see the "Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec" section.

Restrictions for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

- This feature does not support IPv6 traffic.
- This feature does not support transport mode on the Cisco ASR 1000 Series Aggregation Services Routers or on the Cisco VPN Internal Service Module for Cisco Integrated Services Routers Generation 2 (ISR G2).

Information About GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Group Member Registration of Security Group Tagging Capability

When a KS receives a security association (SA) registration request from a group member (GM) or receives a connection establishment request from a cooperative KS, it checks whether any group SA has SGT inline tagging enabled. If so, all GMs and cooperative KSs must register using GET VPN software version 1.0.5 or higher to be accepted. Otherwise, the registration request or establishment request is rejected, and the KS generates a syslog message to notify the network administrator.

Creation of SAs with Security Group Tagging Enabled

After you enable GET VPN support of IPsec inline tagging (using the **tag cts sgt** command) in a group SA and then trigger a rekey (using the **crypto gdoi ks rekey** command), the KS checks for GMs and cooperative KSs in the group not using a compatible software version. If found, a warning message appears:

```
WARNING for group GETVPN: some devices cannot support SGT inline tagging. Rekey can cause
traffic disruption and GM registration failures. Please check 'show crypto gdoi feature
sgt'.
Are you sure you want to proceed ? [yes/no]:
```

Handling of Security Group Tags in the Group Member Data Plane

Egress traffic is traffic sent out from a GDOI-protected interface of a GM. The following table specifies GM behavior for the egress path:

Table 305: Egress Handling of Security Group Tags

Security group tagging is enabled on SA	CTS provides SGTs	GM data plane behavior
Yes	Yes	Adds SGTs to Cisco metadata and encrypts
Yes	No	Encrypts without SGTs

Security group tagging is enabled on SA	CTS provides SGTs	GM data plane behavior
No	Yes	Encrypts without SGTs
No	No	Encrypts without SGTs

Ingress traffic is traffic received by a GDOI-protected interface of a GM. The table below specifies GM behavior for the ingress path:

Table 306: Ingress Handling of Security Group Tags

Security group tagging is enabled on SA	CTS provides SGTs	GM data plane behavior
Yes	Yes	Decrypts and extracts SGTs for CTS
Yes	No	Decrypts without SGT processing
No	Yes	Decrypts and ignores SGTs
No	No	Decrypts without SGT processing

Packet Overhead and Fragmentation When Using Security Group Tagging

Because it adds Cisco metadata containing the SGT information to each GDOI packet, SGT inline tagging increases packet overhead by eight bytes (or 16 bytes with time-based antireplay enabled).

If a packet is fragmented before GDOI encryption, each fragment is inline tagged with SGT information accordingly. If packet is fragmented after GDOI encryption, only the first fragment is inline tagged with SGT information.

You can use two methods to handle fragmentation. The first method is to use the **ip mtu** command on the interface that is handling encryption to accommodate the extra bytes used to carry the SGT information via Cisco metadata. The second method is to use the **ip tcp adjst-mss 1352** command on the GM's LAN interface. This command ensures that the resulting IP packet on the LAN segment is less than 1392 bytes, thereby providing 108 bytes for any overhead plus the Cisco metadata to carry the SGTs.

For more information about designing around MTU issues, refer to the “Designing Around MTU Issues” section of the Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide

How to Configure GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec

You should use the IPsec Inline Tagging for Cisco TrustSec feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature.

Perform this task on the KS (or primary KS) to ensure that all devices in the network support IPsec inline tagging for Cisco TrustSec.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature cts-sgt**
3. **show crypto gdoi feature cts-sgt | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature cts-sgt** **Example:** `Device# show crypto gdoi feature cts-sgt`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports IPsec inline tagging for Cisco TrustSec.
Step 3	**show crypto gdoi feature cts-sgt | include No** **Example:** `Device# show crypto gdoi feature cts-sgt | include No`	(Optional) Displays only those devices that do not support IPsec inline tagging for Cisco TrustSec.

Configuring IPsec Inline Tagging for Cisco TrustSec

To configure IPsec inline tagging for Cisco TrustSec, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **sa ipsec** *sequence-number*
7. **tag cts sgt**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GET-SGT`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.

	Command or Action	Purpose
Step 6	**sa ipsec** *sequence-number* **Example:** `Device(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 7	**tag cts sgt** **Example:** `Device(gdoi-sa-ipsec)# tag cts sgt`	Enables IPsec inline tagging for Cisco TrustSec.
Step 8	**end** **Example:** `Device(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

After enabling IPsec inline tagging, you must trigger a rekey. For more information, see the "Triggering a Rekey" section.

Triggering a Rekey

If you change the security policy (for example, from DES to AES) on the KS (or primary KS) and exit from global configuration mode, a syslog message appears on the KS indicating that the policy has changed and a rekey is needed. You enter the rekey triggering command as described below to send a rekey based on the latest policy in the running configuration.

Perform this task on the KS (or primary KS) to trigger a rekey.

SUMMARY STEPS

1. **enable**
2. **crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**] **Example:** `Device# crypto gdoi ks group mygroup rekey`	Triggers a rekey on all GMs. The optional **replace-now** keyword immediately replaces the old TEKs and KEK on each GM to enable the new policy before the SAs expire. **Note** Using the **replace-now** keyword could cause a temporary traffic discontinuity.

Examples

A message appears on the KS as follows:

```
Device# crypto gdoi ks rekey
Device#
*Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey with
policy-replace for group GET from address 10.0.8.1 with seq # 2
```

After the policy change, when each GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). Each GM continues to encrypt and decrypt traffic using the old SA until its shortened lifetime expires.

If you try to trigger a rekey on the secondary KS, it rejects the command as shown below:

```
Device# crypto gdoi ks rekey
ERROR for group GET: This command must be executed on Pri-KS
```

Verifying and Troubleshooting GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

To view the configuration that is running on a GM, use the **show running-config** command.

To display the number of packets that are tagged with SGTs, enter the following command.

```
Device# show crypto ipsec sa detail

interface: Ethernet0/0
   Crypto map tag: GET, local addr 5.0.0.2
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   Group: GET-SGT
.
.
.
#pkts tagged (send): 0, #pkts untagged (rcv): 5
```

The pkts tagged (send) field displays packets tagged with an SGT in the outbound direction. The pkts untagged (rcv) field displays packets not tagged with an SGT in the inbound direction.

Configuration Examples for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Example: Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support IPsec inline tagging for Cisco TrustSec:

```
Device# show crypto gdoi feature cts-sgt

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.5          Yes
        10.0.6.2         1.0.5          Yes
        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No

    Group Member ID     Version   Feature Supported
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
        10.0.3.1         1.0.5          Yes
        10.0.3.2         1.0.5          Yes
```

You can also enter the above command on a GM (which will display the information for the GM but not for the KS or other GMs).

The following example shows how to enter the command on the KS (or primary KS) find only those devices in the GET VPN network that do *not* support IPsec inline tagging for Cisco TrustSec:

```
Device# show crypto gdoi feature cts-sgt | include No

        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
```

Example: Configuring IPsec Inline Tagging for Cisco TrustSec

The following example shows how to configure CTS SGT inline tagging in an IPsec SA for a KS serving a single GDOI group:

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ACL-SGT
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# crypto gdoi group GET-SGT
```

```
Device(config-gdoi-group)# identity number 1
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# tag cts sgt
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL-SGT
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# end
```

The following example shows how to configure two groups: A group with GMs that are upgraded to GET VPN version 1.0.5 or higher (and therefore supports CTS SGT inline tagging) and a group with GMs that are not yet upgraded. The upgraded GMs will register to group number 1111 (a lower crypto map sequence number) and with group number 2222 (a higher crypto-map sequence number). Non-upgraded GMs will register only to group number 2222.

This example configures SGT tagging for traffic between two sites. The **permit ip** commands add access control entries (ACEs) to the access control list (ACL) that permit communication between the two sites:

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ACL_NET_AB
Device(config-ext-nacl)# permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
Device(config-ext-nacl)# permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
Device(config-ext-nacl)# exit
Device(config)# ip access-list extended ACL_ALL
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# crypto gdoi group GET1
Device(config-gdoi-group)# identity number 1111
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey authentication mypubkey rsa mykey
Device(gdoi-local-server)# rekey transport unicast
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# tag cts sgt
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL_NET_AB
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# exit
Device(gdoi-local-server)# exit
Device(config-gdoi-group)# exit
Device(config)# crypto gdoi group GET2
Device(config-gdoi-group)# crypto gdoi group GET2
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey authentication mypubkey rsa mykey
Device(gdoi-local-server)# rekey transport unicast
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL_ALL
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# end
```

Note GET VPN supports a maximum of 100 ACEs per ACL.

Example: Triggering Rekeys on Group Members

Ensuring That GMs Are Running Software Versions That Support Rekey Triggering

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to display the version of software on devices in the GET VPN network and display whether they support rekey triggering after a policy change:

```
Device# show crypto gdoi feature policy-replace

Key Server ID       Version  Feature Supported
  10.0.8.1         1.0.2          Yes
  10.0.9.1         1.0.2          Yes
  10.0.10.1        1.0.2          Yes
  10.0.11.1        1.0.2          Yes
 Group Member ID     Version  Feature Supported
  5.0.0.2          1.0.2          Yes
  9.0.0.2          1.0.1          No
```

The following example shows how to find only those devices that do not support rekey triggering after policy replacement:

```
Device# show crypto gdoi feature policy-replace  | include No

       9.0.0.2          1.0.1          No
```

For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs.

Triggering a Rekey

The following example shows how to trigger a rekey after you have performed a policy change. In this example, an IPsec policy change (for example, DES to AES) occurs with the **profile gdoi-p2** command:

```
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# no profile gdoi-p
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# end
Device#

*Jan 28 09:15:15.527: %SYS-5-CONFIG_I: Configured from console by console
*Jan 28 09:15:15.527: %GDOI-5-POLICY_CHANGE: GDOI group GET policy has changed. Use
'crypto gdoi ks rekey' to send a rekey, or the changes will be send in the next scheduled
rekey
Device# crypto gdoi ks rekey
Device#
*Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey with
policy-replace for group GET from address 10.0.8.1 with seq # 2
```

The following example shows the error message that appears if you try to trigger a rekey on the secondary KS:

```
Device# crypto gdoi ks rekey

ERROR for group GET: This command must be executed on Pri-KS
```

Note If time-based antireplay (TBAR) is set, the key server periodically sends a rekey to the group members every 2 hours (7200 sec). In the following example, even though the lifetime is set to 8 hours (28800 sec), the rekey timer is set to 2 hours.

```
Device(config)# crypto ipsec profile atm-profile
Device(ipsec-profile)# set security-association lifetime seconds 28800
!
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group ATM-DSL
Device(config-gdoi-group)# server local
Device(gdoi-sa-ipsec)# sa ipsec 1
!
Device(gdoi-sa-ipsec)# replay time window-size 100
```

The commands **show crypto gdoi gm replay** and **show crypto gdoi ks replay** displays TBAR information.

Additional References for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	Cisco IOS GET VPN Solutions Deployment Guide
Configuring Cisco TrustSec	Cisco TrustSec Configuration Guide, Cisco IOS Release 15M&T
Designing around MTU issues	Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide

Standards and RFCs

Standard/RFC	Title
RFC 2401	Security Architecture for the Internet Protocol
RFC 6407	The Group Domain of Interpretation

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 307: Feature Information for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Feature Name	Releases	Feature Information
GET VPN Support of IPsec Inline Tagging for Cisco TrustSec		The Cisco TrustSec (CTS) architecture secures networks by establishing domains of trusted network devices. Once a network device authenticates with the network, the communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and replay protection mechanisms. CTS uses the user and device identification information acquired during the authentication phase to classify packets as they enter the network. CTS maintains classification of each packet or frame by tagging it with a security group tag (SGT) on ingress to the network so that it can be identified for applying security and other policy criteria along the data path. The tags allow network intermediaries such as switches and firewalls to enforce access control policy based on the classification. The GET VPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GET VPN inline tagging to carry the SGT information across the private WAN. The following commands were introduced or modified: **show crypto gdoi, show crypto ipsec sa, tag cts sgt.**

CHAPTER 227

GETVPN GDOI Bypass

The GETVPN GDOI Bypass feature supports enabling and disabling the default Group Domain of Interpretation (GDOI) bypass crypto policy. It also supports hardening of the default GDOI bypass crypto policy once it is enabled.

Restrictions for GETVPN GDOI Bypass

When a key server (KS) is placed behind a group member (GM), the local deny Access Control List (ACL) must be configured explicitly to allow traffic using UDP as the transport protocol and port 848 as either the source or destination (UDP 848 traffic) to pass through.

Information About GETVPN GDOI Bypass

GDOI Bypass Crypto Policy

The Cisco IOS Group Encrypted Transport VPN (GETVPN) uses Group Domain of Interpretation (GDOI) as the key management protocol.

A group member (GM) is a device responsible for encryption and decryption, that is, a device responsible for handling the GET VPN data plane.

A key server (KS) is a device responsible for creating and maintaining the GET VPN control plane. All encryption policies, such as traffic, encryption protocols, security association, rekey timers, and so on, are centrally defined on the KS and are pushed down to all GMs at registration time.

Enabling and Disabling the Default GDOI Bypass Crypto Policy

A new group member (GM) configuration allows users to disable the Group Domain of Interpretation (GDOI) bypass crypto policy and to control traffic exceptions by explicitly configuring the GM local access control list (ACL).

Hardening of the Default GDOI Bypass Crypto Policy

To improve security, the following changes have been enforced while applying the default Group Domain of Interpretation (GDOI) bypass crypto policy:

- The default GDOI bypass crypto policy is installed only on Group Encrypted Transport VPN (GETVPN)-protected interfaces (interfaces at which GDOI crypto map is applied). Only UDP848 traffic that is destined for the group member's (GM) address used for registration or rekey is allowed.
- If the GM VRF-aware feature is used to specify that the GDOI data plane and control plane are in different VRFs, auto-insertion of the default GDOI bypass crypto policy is not applied to the GDOI-protected interface.
- If traffic using UDP as the transport protocol and port 848 as either the source or destination (UDP 848 traffic) is expected to arrive at other non-GDOI-protected interfaces (but with other crypto maps applied), exceptions for the non-GDOI crypto map must be explicitly configured.
- If a crypto map set with multiple groups is configured, the overall GDOI bypass crypto policy installed is the union of all the GDOI bypass crypto policies for each group within the security association database (SADB).

Any of the conditions mentioned below triggers a recompute of the default GDOI bypass crypto policy applied to a GETVPN-protected interface:

- Removing **client bypass-policy** configuration using the **no client bypass-policy** command.
- Applying or removing the GDOI bypass crypto map from an interface.
- Applying or removing the GDOI bypass crypto map from crypto map sets.
- Changing the IP address of the GDOI-protected interface (if **no client registration interface** is used)
    - If **client registration interface** is used, the following cases trigger a recompute of the default GDOI bypass crypto policy applied to a GETVPN-protected interface:
        - Changes from **no client registration interface** to **client registration interface**
        - Changes to the client registration interface (for example, from loopback 0 to loopback 1)
        - Changes to the client registration interface address

How to Configure GETVPN GDOI Bypass

Enabling the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **client bypass-policy**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**client bypass-policy** **Example:** `Device(config-gdoi-group)# client bypass-policy`	Enables the default GDOI bypass crypto policy.
Step 5	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Disabling the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **no client bypass-policy**

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**no client bypass-policy** **Example:** `Device(config-gdoi-group)# no client bypass-policy`	Disables the default GDOI bypass crypto policy.
Step 5	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Verifying Enablement and Disablement of the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi gm acl**
3. **show crypto gdoi gm acl**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show crypto gdoi gm acl**

Verifies the enablement of the default GDOI bypass crypto policy.

Note VRF will be displayed only if it is non-global.

Example:

```
Device# show crypto gdoi gm acl

Group Name: GETVPN
 ACL Downloaded From KS 10.0.0.2:
   access-list   deny eigrp any any
   access-list   permit ip any any
 ACL Configured Locally:
 ACL of default GDOI bypass policy:
   Ethernet1/0: deny udp host 10.0.0.9 eq 848 any eq 848 vrf RED*
```

Step 3 **show crypto gdoi gm acl**

Verifies the disablement of the default GDOI bypass crypto policy.

Example:

```
Device# show crypto gdoi gm acl

Group Name: GETVPN
 ACL Downloaded From KS 10.0.0.2:
   access-list   deny eigrp any any
   access-list   permit ip any any
 ACL Configured Locally:
 ACL of default GDOI bypass policy: Disabled
```

Configuration Examples for GETVPN GDOI Bypass

Example: Enabling the Default GDOI Bypass Crypto Policy

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# client bypass-policy
Device(config-gdoi-group)# end
```

Example: Disabling the Default GDOI Bypass Crypto Policy

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# no client bypass-policy
Device(config-gdoi-group)# end
```

Additional References for GETVPN GDOI Bypass

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GETVPN GDOI Bypass

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 308: Feature Information for GETVPN GDOI Bypass

Feature Name	Releases	Feature Information
GETVPN GDOI Bypass		The following commands were introduced: **client bypass-policy** and **show crypto gdoi gm acl**.

CHAPTER 228

GETVPN G-IKEv2

Cisco Group Encrypted Transport VPN (GET VPN) includes a set of features that are necessary to secure IP multicast group traffic or unicast traffic over an enterprise private WAN that originates on or flows through a Cisco device. The GETVPN G-IKEv2 feature implements Internet Key Exchange version 2 (IKEv2) protocol on GETVPN thereby allowing GETVPN to derive the benefits of IKEv2.

Restrictions for GETVPN G-IKEv2

- You can configure either Group Key Management (GKM) or Group Domain of Interpretation (GDOI) for a group member (GM), whereas you can configure both GKM and GDOI for a key server (KS).
- IKEv2 for COOP is not supported. Use IKEv1 for COOP between the key servers in the G-IKEv2 setup.
- EAP is not currently supported with G-IKEv2.
- GETVPN G-IKEv2 does not support IP-D3P. IP-D3P with G-IKEv2 is yet to be supported on GETVPN Group Members (GMs).

Information About GETVPN G-IKEv2

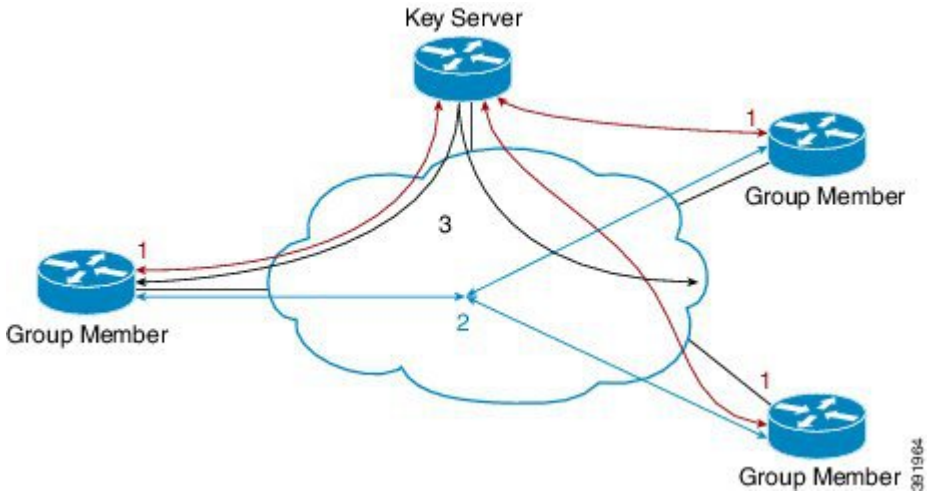
Overview of GETVPN G-IKEv2

Cisco Group Encrypted Transport Virtual Private Network (GETVPN) architecture is based on the Group Domain of Interpretation (GDOI) protocol. GETVPN uses Internet Security Exchange and Key Management Protocol (ISAKMP) to authenticate new group members, download cryptographic policy, and distribute traffic encryption key (TEK) and key encryption key (KEK) to group members. However, Internet Key Exchange Version 2 (IKEv2) has replaced. IKEv2 reduces network latency, reduces complexity in message exchanges, improves interoperability and reliability, and fixes cryptographic issue in HASH authentication. GET VPN combines IKEv2 protocol with IPsec to provide an efficient method to secure IP multicast traffic or unicast

traffic through the GETVPN G-IKEv2 feature. This feature provides a complete IKEv2 solution across all of Cisco's VPN technologies.

The G-IKEv2 protocol provides a mechanism for a group member (GM) to download policy and keys from a key server (KS). These policy and keys are used to secure communication among GMs in a group. G-IKEv2 is a new model to secure group communication between remote locations in an enterprise private WAN. The following figure depicts the basic system architecture of GETVPN using G-IKEv2 to register GM's with a KS and download keys and policy to GM's from a KS.

Figure 127: GETVPN Architecture through G-IKEv2 Protocol



Internet Key Exchange Version 2 (IKEv2)

Internet Key Exchange Version 2 (IKEv2), a next-generation key management protocol based on RFC 4306, is an enhancement of the IKE Protocol. IKEv2 is used for performing mutual authentication and establishing and maintaining security associations (SAs). For more information on IKEv2, see *FlexVPN and Internet Key Exchange Version 2 Configuration Guide*.

The following table compares the tunnel performance between IKE and IKEv2.

Protocol	Tunnels per Second	Maximum Simultaneous Tunnels
IKE	45	60
IKEv2	89	200

The benefits of IKEv2 are as follows:

Dead Peer Detection and Network Address Translation-Traversal

Internet Key Exchange Version 2 (IKEv2) provides built-in support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).

Certificate URLs

Certificates can be referenced through a URL and hash, instead of being sent within IKEv2 packets, to avoid fragmentation.

Denial of Service Attack Resilience

IKEv2 does not process a request until it determines the requester, which addresses to some extent the Denial of Service (DoS) problems in IKEv1, which can be spoofed into performing substantial cryptographic (expensive) processing from false locations.

EAP Support

IKEv2 allows the use of Extensible Authentication Protocol (EAP) for authentication.

Multiple Crypto Engines

If your network has both IPv4 and IPv6 traffic and you have multiple crypto engines, choose one of the following configuration options:

- One engine handles IPv4 traffic and the other engine handles IPv6 traffic.
- One engine handles both IPv4 and IPv6 traffic.

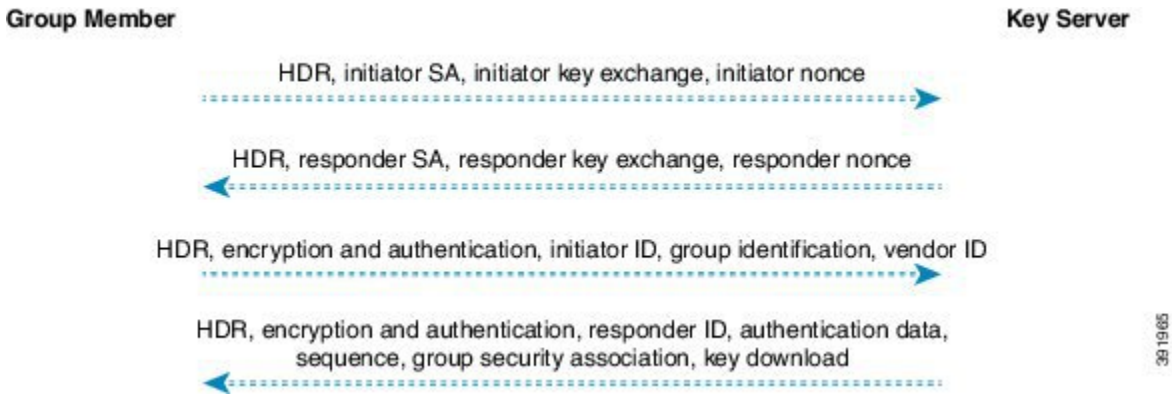
Reliability and State Management (Windowing)

IKEv2 uses sequence numbers and acknowledgments to provide reliability, and mandates some error-processing logistics and shared state management.

GETVPN G-IKEv2 Exchanges

The message exchanges between GM and KS conforms to the Internet Engineering Task Force (IETF) Group Key Management using IKEv2 Standards draft.

Figure 128: G-IKEv2 Message Exchanges



1. Group member initiates a registration request to key server by sending preferred cryptographic algorithms (in SAi payload), Diffie–Hellman public number, in initiator's key exchange (KE) phase 1 payload, and nonce, which is a random number for guaranteeing liveness in Initiator's nonce payload.
2. Key server responds with the negotiated cryptographic algorithm (in responder's SA phase 1 payload), Diffie–Hellman public number (in responder's KE payload), nonce (in responder's nonce payload). Optionally, if key server is configured to use Rivest, Shamir, and Adleman (RSA) digital signature as an authentication method, key server also sends a certificate request.

3. On receiving key server's response to the registration request, the group member uses the cryptographic algorithm in the SAr1 payload and Diffie–Hellman value to create keys and to encrypt the message sent to the key server. The encrypted message includes the initiator's ID and, optionally, certificate and certificate request, if RSA digital signature is used as authentication method. In case of Suite B implementations, a notify payload is sent for requesting sender IDs used with Galois/Counter Mode (GCM)–Advanced Encryption Standard (AES) or Galois Message Authentication Code (GMAC)–Advanced Encryption Standard (AES) transforms.

Note Group member requests a set of sender IDs applicable for interfaces for a lifetime of one day. After receiving the lifetime in a registration (for Long SA Lifetime) or a rekey (for Short SA Lifetime) message, group member stores the lifetime for calculating the number of sender IDs for future registrations.

4. After authenticating group manager, key server authorizes group member before registering group manager. After registration, key server sends the group's policy (in the GSA payload) and the group's keying material (in the KD payload) to group manager. The SEQ payload is optional and is sent when the key server wants to inform group manager of the current sequence number of the rekey message. These payloads are included in the GSA_AUTH response message.

Group Member Communication

Group members do not establish IPsec tunnels with one another, but use the IPsec policy and keys to secure communication between group members in a group.

Future Registrations

When a secure registration channel is established between group manager and key server, additional group member registrations for additional groups occurs through the established secure registration channel. In such scenario, group member uses the GSA_CLIENT_SERVER exchange that includes the group ID (IDg) to request either key encryption keys (KEK) or traffic encryption keys (TEKs) or both from key server.

Key Server Rekey

Key server distributes new group keys to group members using the G-IKEv2 group maintenance channel via unicast or multicast communication. Rekey is optional in G-IKEv2. When rekey is used, the KS sends a rekey message to group member. This message could be unicast or multicast depending on the key server configuration. Key server uses the KEK that is sent to the group member during registration to encrypt the rekey message. On receiving a rekey message, group member must ensure that the SEQ number in the rekey message is larger than the last received SEQ number. Group member could have received the SEQ number either via a registration message or a rekey message, whichever is later. If key server group is configured as both GDOI (IKEv1) and G-IKEv2 group, two rekey messages are sent—one over GDOI and another over G-IKEv2—for multicast rekey. In case of unicast rekey, key server only sends a GDOI or G-IKEv2 rekey depending on the group member's mode or type.

Note If the rekey is unicast, the group member must send an acknowledgment to key server.

Supported Features and GKM Version

The GETVPN G-IKEv2 feature supports the existing GETVPN features, which are as follows:

- Rekey and retransmission
- GM access control list (ACL)
- Fail-close mode
- Receive-only mode
- Anti-replay
- Authentication policy for group member registration
- GDOI MIBS
- VRF-Aware group member
- Group member removal and policy replacement
- Cooperative key server
- GETVPN IPv6 dataplane
- IPsec inline tagging support
- GETVPN resiliency phase 1 and phase 2
- Cooperative announcement message optimization

The GETVPN G-IKEv2 feature is supported in GKM version 1.0.12 and later releases. The supported GKM versions for a key server is 1.0.13 and a group member is 1.0.12. The difference between versions on a key server and a group member is because the IP D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server features are available on the key server from 1.0.13 only.

GDOI to G-IKEv2 Migration

Over a period of time, you may want to upgrade and migrate your key servers and group members to G-IKEv2. Migration from GDOI to G-IKEv2 for an entire GETVPN group requires careful planning. You cannot migrate all your group members at the same time. The migration entails allowing GDOI group members and G-IKEv2 group members to communicate using the same traffic encryption key (TEK) while using different control plane protocols—GDOI and G-IKEv2. A GDOI to G-IKEv2 migration sequence includes the following:

- Backward compatibility—The new Cisco IOS software image containing the GETVPN G-IKEv2 feature must support existing GDOI features and must be consistent with for earlier releases of GDOI features for Cisco IOS software.
- Service upgrade—The recommended sequence for changing the Cisco IOS software image is secondary key server, primary key server, and group member.
- Service downgrade—The recommended sequence for changing the Cisco IOS software image is group member, secondary key server, and primary key server.

Service Upgrade Procedure

1. Save the existing key server and group member GDOI configurations. For more information, see the "Configuration Replace and Configuration Rollback" feature module in the *Managing Configuration Files Configuration Guide*.
2. Configure a key encryption key (KEK) and a traffic encryption key (TEK) lifetime on all key servers to avoid network split and merge during the migration of the key servers. Use the crypto gdoi ks rekey command to configure the new lifetimes.
3. Upgrade key server to the new Cisco IOS software images. Follow the sequence mentioned above—start with the secondary key server followed by the primary key server. All existing configurations that use the keyword **gdoi** will be converted to the keyword **gkm**. For example, the global configuration command **crypto gdoi group** will be converted to **crypto gkm group** command. However, the groups continue to use GDOI for registration and rekey.
4. On key server, execute the **gikev2** command in the server local command for groups that support GDOI and G-IKEv2 group members.
5. Upgrade group members to the new Cisco IOS software image. All existing configurations that use the keyword "**gdoi**" will be converted to the keyword **gkm**. For example, the global configuration commands **crypto gdoi group** and **crypto map gdoi** will be converted to "**crypto gkm group**" and **crypto map gkm** respectively. These groups continue to use GDOI for registration and rekey and include the **client protocol gdoi** command.
6. Configure the **client protocol gikev2** command to use G-IKEv2 on group member.
7. Configure the **no gdoi** command in the server local command, to stop servicing GDOI group members.

For a group member to use GDOI after upgrading to G-IKEv2, configure the **client protocol gdoi** command in the group member group configuration. Group member registers again with key server using GDOI instead of G-IKEv2.

Note Before you convert group member, ensure that key server to which group member is registered is configured with the gdoi command in GDOI local server configuration mode.

Service Downgrade Procedure

Use the previously saved GDOI configurations (saved before upgrade procedure) and downgrade the Cisco IOS software for each group member. Next, downgrade the key server; beginning with the secondary key server followed by primary key server. For more information, see the "Configuration Replace and Configuration Rollback" feature module in the *Managing Configuration Files Configuration Guide*.

Migration Examples

This section provides examples on GDOI to G-IKEv2 migration. The following examples show how the GDOI group g1 is converted to a GKM group after upgrading to a G-IKEv2 Cisco IOS software image. The following is a sample key server configuration before Cisco IOS software upgrade.

```
crypto gdoi group g1
 identity 1111
 server local
  .
```

```
  .
  .
  sa ipsec 1
   profile getvpn_profile
   match address getvpn_acl
   .
   .
   .
   redundancy
   .
   .
   .
```

The following is a sample key server configuration after Cisco IOS software upgrade. In this example, the commands **gdoi**, **no gikev2**, and **gikev2** are automatically added. The **gikev2** command starts accepting G-IKEv2 registrations.

```
crypto gkm group g1
 identity 1111
 server local
  gdoi
  no gikev2
  gikev2 ikev2_profile1
  .
  .
  .
  sa ipsec 1
   profile getvpn_profile
   match address getvpn_acl
   .
   .
   .
   redundancy
   .
   .
   .
```

The following is a sample group member configuration before Cisco IOS software upgrade.

```
crypto gdoi group g1
 identity 1111
 server address ipv4 ks1
 server address ipv4 ks2

crypto map GETVPN_CM 10 gdoi
 set group g1

interface g0/0/0
 crypto map GETVPN_CM
```

The following is a sample group member configuration after Cisco IOS software upgrade. In this example, the commands **client protocol gdoi** and **client protocol gikev2** are automatically added. The **client protocol gikev2** command starts using G-IKEv2.

```
crypto gkm group g1
 identity 1111
 server address ipv4 ks1
 server address ipv4 ks2
 client protocol gdoi
 client protocol gikev2 ikev2_profile1 ] – Configure this to start using G-IKEv2

crypto map GETVPN_CM 10 gdoi
 set group g1
```

```
interface g0/0/0
 crypto map GETVPN_CM
```

GETVPN G-IKEv2 Configuration

All GETVPN commands—EXEC and global configuration commands—include the keyword **gdoi**. G-IKEv2 does not include the Domain of Interpretation, therefore, a generic abbreviation **gkm** referring to Group Key Management is used for a group that can use either GDOI or G-IKEv2 protocols for registration and rekey. As of now, both commands **crypto gdoi** and **crypto gkm** are available. However, the **GDOI** keyword will be deprecated and replaced by the **gkm** keyword in future. For example, to configure a key server group, the GDOI command is **crypto gdoi group** *group-name*, whereas the GKM command would be **crypto gkm group** *group-name*.

How to Configure GETVPN G-IKEv2

Configuring an IKEv2 Profile

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}
5. **identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*}
6. **keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] }
7. **match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*}
8. **pki trustpoint** *trustpoint-label* [**sign** | **verify**]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile gkm-gikev2`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}} **Example:** `Device(config-ikev2-profile)# authentication local ecdsa-sig`	Specifies the local or remote authentication method. • **rsa-sig**—Specifies RSA-sig as the authentication method. • **pre-share**—Specifies the preshared key as the authentication method. • **ecdsa-sig**—Specifies ECDSA-sig as the authentication method. • **eap**—Specifies EAP as the remote authentication method. • **query-identity**—Queries the EAP identity from the peer. • **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response. **Note** You can specify only one local authentication method but multiple remote authentication methods.
Step 5	**identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# identity local email abc@example.com`	This is an optional step. Specifies the local IKEv2 identity type. **Note** If the local authentication method is a preshared key, the default local identity is the IP address. If the local authentication method is a Rivest, Shamir, and Adleman (RSA) signature, the default local identity is a Distinguished Name.
Step 6	**keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] } **Example:** `Device(config-ikev2-profile)# keyring aaa keyring1 name-mangler mangler1`	Specifies the local or AAA-based key ring that must be used with the local and remote preshared key authentication method. **Note** You can specify only one key ring. Local AAA is not supported for AAA-based preshared keys. **Note** Depending on your release, the **local** keyword and the **name-mangler** *mangler-name* keyword-argument pair should be used.

	Command or Action	Purpose
		Note When using AAA, the default password for a Radius access request is "cisco". You can use the **password** keyword within the **keyring** command to change the password.
Step 7	**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# match address local interface Ethernet 2/0`	Uses match statements to select an IKEv2 profile for a peer.
Step 8	**pki trustpoint** *trustpoint-label* [**sign** | **verify**] **Example:** `Device(config-ikev2-profile)# pki trustpoint tsp1 sign`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method. **Note** If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification. **Note** In contrast to IKEv1, a trustpoint must be configured in an IKEv2 profile for certificate-based authentication to succeed. There is no fallback for globally configured trustpoints if this command is not present in the configuration. The trustpoint configuration applies to the IKEv2 initiator and responder.
Step 9	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits the IKEv2 profile configuration mode and returns to the privileged EXEC mode.

Configuring GKM Policy on a Key Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group** [**ipv6**] *group-name*
4. **server local**
5. **gikev2** *IKEv2-profile-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gkm group gkm-grp1`	Configures a GKM policy and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates a device as a GKM key server and enters GKM local server configuration mode.
Step 5	**gikev2** *IKEv2-profile-name* **Example:** `Device(gkm-local-server)# gikev2 gkm-gikev2`	Enables G-IKEv2 profile for registration and rekey on a key server.
Step 6	**end** **Example:** `Device(gkm-local-server)# end`	Exits GKM local server configuration mode and returns to privileged EXEC mode.

Configuring GKM Policy on Group Member

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group** [**ipv6**] *group-name*
4. **client protocol gikev2** *gkm-gikev2*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gkm group gkm-grp2`	Configures a GKM policy and enters GKM group configuration mode.
Step 4	**client protocol gikev2** *gkm-gikev2* **Example:** `Device(config-gkm-group)# client protocol gikev2 gkm-gikev2`	Enables G-IKEv2 profile for registration and rekey on a group member.
Step 5	**end** **Example:** `Device(config-gkm-group)# end`	Exits GKM group configuration mode and returns to privileged EXEC mode.

Additional References for GETVPN G-IKEv2

Related Documents

Related Topic	Document Title
Security Commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*

Standards and RFCs

Standard/RFC	Title
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
Group Key Management using IKEv2	*draft-yeung-g-ikev2-07*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GETVPN G-IKEv2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 309: Feature Information for GETVPN G-IKEv2

Feature Name	Releases	Feature Information
GETVPN G-IKEv2		The following commands were introduced or modified: **client protocol**, **crypto gkm group**, **gikev2**, **show crypto gkm**.

CHAPTER 229

8K GM Scale Improvement

The 8K GM Scale Improvement feature supports optimization of the Cooperative Protocol (COOP) announcement messages by increasing the number of Group Members (GM) to 8000.

Prerequisites for 8K GM Scale Improvement

To upgrade or downgrade a particular protocol version, maintain the same policies, keys, and group member (GM) database to ensure uninterrupted communication between GMs.

Information About 8K GM Scale Improvement

8K GM Scale Improvement

A Cooperative Protocol Announcement (COOP ANN) message has several clients and each client is associated with a protocol version. The COOP ANN message has been optimized to hold up to 8000 Group Members (GM), subsequently increasing the protocol version of the GM header.

This feature also supports upgrade and downgrade of a GM header protocol version.

How to Configure 8K GM Scale Improvement

Upgrading and Downgrading the Group Member Header Protocol Version

Before you begin

- Ensure that all Key Servers (KS) are upgraded to the “optimize” protocol version before scaling the network to more than 4000 GMs
- Ensure that all upgraded KSs must be downgraded to the “base” protocol version before scaling down to a network that supports only up to 4000 GMs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **server local**
5. **redundancy**
6. **protocol version** {**base** | **optimize**}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Identifies a group server defined locally and enters GDOI local server configuration mode.
Step 5	**redundancy** **Example:** `Device(gdoi-local-server)# redundancy`	Enters GDOI COOP KS configuration mode. **Note** Ensure that the local server source address is defined.

	Command or Action	Purpose
Step 6	**protocol version** {**base** | **optimize**} **Example:** `Device(gdoi-coop-ks-config)# protocol version optimize`	Upgrades or downgrades the protocol version of the GM header. • **base**—COOP ANN message supports up to 4000 GMs. • **optimize**—COOP ANN message supports up to 8000 GMs.
Step 7	**end** **Example:** `Device(gdoi-coop-ks-config)# end`	Exits COOP KS configuration mode and returns to privileged EXEC mode.

Configuration Examples for 8K GM Scale Improvement

Example: Upgrading the Group Member Header Protocol Version

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# redundancy
Device(gdoi-coop-ks-config)# protocol version optimize
Device(gdoi-coop-ks-config)# end
```

Example: Downgrading the Group Member Header Protocol Version

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# redundancy
Device(gdoi-coop-ks-config)# protocol version base
Device(gdoi-coop-ks-config)# end
```

IPSEC Encryption and Decryption in GETVPN

In GETVPN IPsec flow, inbound traffic decryption might not happen in the expected IPsec flow recorder. The decrypted traffic can be recorded in any IPsec SA, if available. The decryption might happen in a random IPsec flow recorder. The following is an example:

```
Device# ping vrf cust1 48.1.1.1 so 38.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 48.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 38.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Device# show crypto session ivrf cust1  detail | sec permit ip 38.0.0.0
  IPSEC FLOW: permit ip 38.0.0.0/255.0.0.0 48.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins
        Outbound: #pkts enc'ed 5 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins

Device# show crypto session ivrf cust1  detail | sec permit ip 48.0.0.0
  IPSEC FLOW: permit ip 48.0.0.0/255.0.0.0 38.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins

Device# show crypto session ivrf cust1  detail | sec permit ip 45.0.0.0
  IPSEC FLOW: permit ip 45.0.0.0/255.0.0.0 35.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 5 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 15
mins
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 15
mins
```

In the above example, flow inbound traffic is not decrypted in the expected IPsec flow.

To overcome this issue and view the number of encrypted and decrypted packets, you can use the following **show** command. Here's a sample output of the **show** command.

```
Device# show crypto gdoi group v6-cust-gdoi1 gm dataplane counters

Data-plane statistics for group v6-cust-gdoi1:
   #pkts encrypt            : 1912   #pkts decrypt            : 1914
    #pkts tagged (send)      : 1841     #pkts untagged (rcv)     : 1834
    #pkts no sa (send)       : 0        #pkts invalid sa (rcv)   : 0
    #pkts encaps fail (send) : 0        #pkts decap fail (rcv)   : 0
    #pkts invalid prot (rcv) : 0        #pkts verify fail (rcv)  : 0
    #pkts not tagged (send)  : 0        #pkts not untagged (rcv) : 0
    #pkts internal err (send): 0        #pkts internal err (rcv) : 0
```

Additional References for 8K GM Scale Improvement

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 310: Feature Information

Feature Name	Releases	Feature Information
8K GM Scale Improvement		The 8K GM Scale Improvement feature supports optimization of the Cooperative Protocol (COOP) announcement messages by increasing the number of Group Members (GM) to 8000. The following command was modified: **protocol**.

CHAPTER 230

GET VPN Interoperability

The D3P Support on GETVPN Key Server, Activation Time Delay, and GDOI Interop ACK for Cisco GETVPN Key Server features enhance interoperability between key servers and group members.

Prerequisites for GET VPN Interoperability

- To enable the feature for a group, ensure that all the devices in a group are running compatible Cisco IOS software and Group Domain of Interpretation (GDOI) versions.
- Enable the Unicast Rekey functionality on a GDOI group before configuring the Internet-Draft ACK for Cisco GETVPN Key Server and Activation Time Delay features.

Restrictions for GET VPN Interoperability

- The IP-D3P support on GETVPN Key Server feature cannot coexist with the GETVPN Resiliency - GM Error Detection and GET VPN Support of IPsec Inline Tagging for Cisco TrustSec features. The latter features must be disabled before enabling IP-D3P support on a GET VPN key server and the IP-D3P must be disabled before enabling GETVPN Resiliency support on GETVPN Key Server.
- The Activation Time Delay feature supports only on IPsec security association. Multiple IPsec SA must not be configured.
- Cisco-Metdata and IP-D3P cannot coexist. When switching between CMD-feature and IP-D3P, the keyserver must perform **crypto gdoi ks rekey replace** to all the GMs to make sure these two features are not enabled simultaneously.
- ASR1K supports IP-D3P only in GETVPN IPv4 tunnel mode.

Information About GET VPN Interoperability

Overview of IP-Delivery Delay Detection Protocol (IP-D3P)

IP datagrams can be subject to a delivery delay attack, where a host or gateway receives datagrams that are not fresh. A fresh datagram is defined as a "Recently generated; not replayed from some earlier interaction of the protocol." An IP-D3P datagram consists of a header and an IP payload. The IP-D3P header includes a timestamp that is used by the receivers of the packet to determine if the packet has been recently generated. Receivers compare the timestamp delivered in the IP packet to their local time and thus determine whether the packet should be accepted.

IP-D3P uses the system clock of group members to create and verify the IP-D3P datagram's timestamp. In most cases, the system clock is set from an external protocol, such as Network Time Protocol (NTP) to synchronize the system clocks of the sender and receiver.

The D3P support on GETVPN Key Server feature enables support for IP-D3P on GET VPN.

IP-D3P Support for Key Server

A new configuration command, **d3p**, in the GDOI local server configuration mode allows you to enable IP-D3P on a key server. After you enable the D3P command, the primary key server issues a rekey to all the group members having a Group Associated Policy (GAP) payload with D3P attributes. The GAP payload includes the following attributes in the rekey message:

- D3P-TYPE—Portable Operating System Interface (POSIX) time, in milliseconds.
- D3P-WINDOWSIZE—IP-D3P window size, in milliseconds.

The **show crypto gkm ks** command displays the IP-D3P parameters that are enabled on a key server.

IP-D3P Support for Cooperative Key Server

If a GET VPN group has more than one key server, IP-D3P must be enabled on all the key servers. The primary key server sends the GAP payload containing the IP-D3P attributes to the secondary key servers through an announcement message, which notifies all cooperative key servers that IP-D3P is now enforced in the group.

On receiving the GAP payload, cooperative key servers check the IP-D3P attributes against their group configuration. If there is a mismatch, cooperative key servers generate a syslog message, warning the network administrator of a misconfiguration or incorrect configuration, as:

```
%GDOI-3-COOP_CONFIG_MISMATCH: WARNING: IP-D3P configuration between Primary KS and Secondary
 KS are mismatched
```

IP-D3P Support for Group Member

Group members receive the IP-D3P parameters present in the rekey messages. Group members process the new GAP payload attributes—D3P-TYPE and D3P-WINDOWSIZE. The window-size, which must be used in IP-D3P for a group member, can be overwritten by using the **client d3p** command in the GDOI group configuration. For example, if a key server configuration is **d3p window msec 1000** and a group member configuration is **client d3p window sec 50**, the group member can enable IP-D3P using the following parameters and overriding the parameters received from thekey server:

```
D3P-TYPE = POSIX-TIME-MSEC
D3P-WINDOWSIZE = 50000
```

Use the **show crypto gdoi gm** command to display the IP-D3P configuration of a group member and the IP-D3P errors, if any, that were encountered.

Note IP-D3P cannot be enabled on Cisco ASR 9000 Series Aggregation Services Routers, which use the parameters sent by a key server. Use the **show crypto gdoi group** command to view the parameters sent by the key server on Cisco ASR 9000 Series Aggregation Services Routers.

Activation Time Delay

GET VPN supports the Activation Time Delay (ATD) feature, in which a key server instructs group members to delay the use of new security associations (SAs) for traffic encryption. A key server includes the ATD value in the Group Associated Policy (GAP) payload when sending unicast rekey messages to group members. The time delay value is not user configurable; it is fixed as 30 seconds before SA expiry. The formula for calculating the ATD value is as follows:

ATD = Max((Max(old-SA-remaining-lifetime_sec, 30sec) – 30sec), 1sec)

Note ATD support is limited to group members that are configured on Cisco ASR 9000 Series Aggregation Services Routers and on non-Cisco devices. Therefore, a key server does not send ATD information to devices other than Cisco ASR 9000 Series Aggregation Services Routers and non-Cisco devices

.

Rekey Acknowledgment

When a key server sends a rekey message to group members for updating the keys and policies of a group, it is useful for a key server to know if all group members have received the rekey message and have successfully processed, installed, and responded to the new keys and policies.

Cisco Unicast Rekey Acknowledgment Message

If a unicast rekey is configured, a key server sends rekey messages, for which group members reciprocate by sending an acknowledgment rekey message.

Note There is no acknowledgment message if multicast rekey is configured.

If a key server sends three consecutive unacknowledged unicast rekeys to a group member, and if the unicast rekeys are unacknowledged by that group member, the group member is removed from the group member database in the key server and no further unicast rekeys are sent to that group member.

GDOI I-D Rekey Acknowledgement Message

The GDOI Interop ACK for Cisco Key Server feature implements the standards for rekey acknowledgment messages between non-Cisco group members and a key server, as defined in the RFC-8263, GROUPKEY-PUSH Acknowledgment message.

The GDOI GROUPKEY-PUSH Acknowledgment message, which is referred to as GDOI I-D Rekey ACK, differs from the Cisco unicast rekey acknowledgment message by defining an interoperable method for a group member to send a rekey acknowledgment to any key server in a group.

GDOI I-D Rekey ACK Support for a Key Server

The **rekey acknowledgement** command enables the key server to request group members to acknowledge rekeys depending on the keywords chosen with the command:

- **cisco**—Accepts Cisco-proprietary rekey ACK (encrypted) message.
- **interoperable**—Requests and accepts rekey ACK (unencrypted) message as per the corresponding Internet Draft.
- **any**—Accepts any supported ACK message based on the group key member version.

After enabling the **rekey acknowledgement** command, the key server sends a new policy attribute, **KEK_ACK_REQUESTED**. The new policy attribute in the key encryption key (KEK) SA payload for registration and rekey.

GDOI I-D Rekey ACK Support for Cooperative Key Server

The **rekey acknowledgement** command must be configured on all the key servers if a GET VPN group has multiple key servers. When a primary key server sends an announcement message to a secondary key server, the primary key server also includes the **KEK SA** payload carrying the **KEK_ACK_REQUESTED** attribute. This notifies all the cooperative key servers to send the **KEK_ACK_REQUESTED** attribute to the group members registered under them.

Upon receiving the **KEK SA** payload with the **KEK_ACK_REQUESTED** attribute, cooperative key servers check their group configuration. If there is a mismatch, cooperative key servers generate a message, warning the network administrator of a misconfiguration or incorrect configuration, as shown here:

```
%GDOI-3-COOP_CONFIG_MISMATCH: WARNING: Interoperable Rekey ACK configuration between Primary
 KS and Secondary KS are mismatched
```

Note Rekey acknowledgments are sent only to a primary key server because it is the primary key server that sends rekey messages. A rekey acknowledgment is sent to a cooperative server only when a cooperative key server is promoted as a primary key server, and if the old primary key server did not create a key encryption key (KEK) or traffic encryption key (TEK) policy.

GDOI I-D Rekey Support for Group Member

A group member is said to support the Internet-Draft ACK for Cisco GETVPN Key Server feature if the group member receives the rekey message containing the KEK_ACK_REQUESTED attribute in the KEK SA payload and sends the GDOI I-D Rekey ACK to the key server through an acknowledgment message.

Key Server and Group Member Communication

When a key server sends the KEK_ACK_REQUESTED attribute in the KEK SA payload, a group member must respond to subsequent rekey messages with the GDOI I-D Rekey ACK unless notified otherwise by the corresponding key server. The communication between a key server and group members are as follows:

1. For every GROUPKEY-PUSH message sent by a key server, the group member must respond with the GROUP-PUSH-KEY ACK message.
2. The key server verifies and validates the message for format and payload. If validation fails, the message is dropped.
3. If validation is successful, the key server processes the SEQ and ID payloads to record the latest acknowledged sequence number for the group member associated with the ID. The sequence number must be the same as the last sent sequence number; otherwise, the SEQ and ID payload will not be recorded.

Note In case of a Cisco key server, a group member is removed from the database if a group member does not send an acknowledgment for three consecutive rekey messages. If a group member is configured with the unicast rekey feature and the KEK_ACK_REQUESTED attribute is not sent for a given KEK Security Parameter Index (SPI), the group members must send the Cisco Unicast Rekey ACK message to the key server.

The following table explains the attributes sent in the KEK SA payload along with the values sent for each acknowledgment option configured on a key server:

Table 311: KEK SA Payload for Each Acknowledgment Option

Acknowledgement Option	New Cisco Group Member	Cisco ASR 9000 Group Member	Non-Cisco Group Member
Cisco	No Attribute	No Attribute	No Attribute
Interoperable	K E K _ A C K _ R E Q REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256
Any	No Attribute	KEK_ACK_REQ REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256

Note When the **no rekey acknowledgement** command is used to set the rekey acknowledgment to the default value ‘Cisco’, the key server does not include the KEK_ACK_REQUESTED attribute in the KEK SA payload.

The following table explains the acknowledgment methodology for each acknowledgment type configured via the keywords in the **rekey acknowledgement** command on a key server:

Table 312: Acknowledgment Methodology

Acknowledgement Option	Key Server Accepts I-D ACK	Key Server Accepts Cisco ACK
Cisco	No (results in error)	Yes
Interoperable	Yes	No (results in error)
Any	Yes	Yes

How to Configure GET VPN Interoperability

Ensuring the Correct GDOI Version on a Key Server

SUMMARY STEPS

1. **enable**
2. **show crypto gkm feature** *feature name*
3. **show crypto gkm feature** *feature-name* | **include no**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

```
Device> enable
```

Step 2 **show crypto gkm feature** *feature name*

Displays the GDOI version running on each key server and group member in the network and information about whether the device supports GET VPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

Example:

```
Device# show crypto gkm feature ip-d3p
Group Name: GET VPN1
    Key Server ID       Version   Feature Supported
        10.0.8.1         1.0.11         Yes
        10.0.9.1         1.0.10         No
    Group Member ID     Version   Feature Supported
        10.0.3.1         1.0.11         Yes
        10.65.9.2        1.0.10         No
```

Example:

```
Device# show crypto gkm feature gdoi-interop-ack
Group Name: GET VPN2
    Key Server ID       Version   Feature Supported
        10.0.8.1         1.0.11         Yes
        10.0.9.1         1.0.10         No
    Group Member ID     Version   Feature Supported
        10.0.3.1         1.0.11         Yes
        10.65.9.2        1.0.10         No
```

Step 3 **show crypto gkm feature** *feature-name* | **include no**

(Optional) Finds devices that do not support a feature.

Example:

```
Device# show crypto gkm feature gdoi-interop-ack | include no
```

Ensuring the Correct GDOI Version on a Group Member

SUMMARY STEPS

1. **enable**
2. **show crypto gkm feature** *feature name*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show crypto gkm feature** *feature name*

Displays the GDOI version running on a group member in the network and information about whether the device supports GET VPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

Example:

```
Device# show crypto gkm feature ip-d3p
       Version    Feature Supported
        1.0.11           Yes
```

Example:

```
Device# show crypto gkm feature gdoi-interop-ack
       Version    Feature Supported
        1.0.10           No
```

Enabling IP-D3P on a Key Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GETVPN**
4. **server local**
5. **sa d3p window**{**sec** *seconds* | **msec** *milliseconds*}
6. **exit**
7. **show crypto gkm ks replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GETVPN** **Example:** `Device(config)# crypto gkm group GETVPN`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates the device as a key server and enters GDOI local server configuration mode.
Step 5	**sa d3p window** {**sec** *seconds* | **msec** *milliseconds*} **Example:** `Device(gdoi-local-server)# sa d3p window msec 5000`	Enables IP delivery delay detection protocol (IP-D3P) on all security associations in the group. • **sec** *seconds*—Window size, in seconds. The range is from1 to 100. • **msec** *milliseconds*—Window size, in milliseconds. The range is from 100 to 10000.
Step 6	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 7	**show crypto gkm ks replay** **Example:** `Device# show crypto gkm ks replay`	Displays key server group information for time-based anti-replay.

Example

The following is a sample output from the **show crypto gkm ks replay** command:

```
Device# show crypto gkm ks replay
Anti-replay Information For Group GETVPN:
    IP-D3P: Type = POSIX-TIME-MSEC, Window-size = 5000 msec
```

Enabling IP-D3P on a Group Member

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GET**
4. **client d3p window** {**sec** *seconds* | **msec** *milliseconds*}
5. **exit**
6. **show crypto gkm gm replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GET** **Example:** `Device(config)# crypto gkm group GETVPN`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**client d3p window** {**sec** *seconds* | **msec** *milliseconds*} **Example:** `Device(config-gkm-group)# client d3p window sec 50`	Enables client-acceptable IP delivery delay detection protocol (IP-D3P). • **sec** *seconds*—Window size, in seconds. The range is from1 to 100. • **msec** *milliseconds*—Window size, in milliseconds. The range is from 100 to 10000.
Step 5	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 6	**show crypto gkm gm replay** **Example:** `Device# show crypto gkm gm replay`	Displays group member information for time-based anti-replay.

Example

The following is a sample output from the **show crypto gkm gm replay** command:

```
Device# show crypto gkm gm replay
Anti-replay Information For Group GET:
   IP-D3P:
     Posix-time-msec           : 502764.17
     Input Packets             : 5        Output Packets           : 5
     Input Error Packets       : 5        Output Error Packets     : 0

IP-D3P Error History (sampled at 10pak/min):
     xx:xx:xx.xxx PST Tue Feb 25 2014: src=5.0.0.2; my_time=502729.95; peer_time=33.46;
win=10
     yy:yy:yy.yyy PST Tue Feb 25 2014: src=5.0.0.2; my_time=502723.95; peer_time=27.45;
win=10
```

Enabling Rekey Acknowledgment

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GET**
4. **server local**
5. **rekey acknowledgement**{**cisco** | **interoperable** | **any**}
6. **exit**
7. **show crypto gkm ks replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GET** **Example:** `Device(config)# crypto gkm group GET`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates the device as a key server and enters GDOI local server configuration mode.
Step 5	**rekey acknowledgement**{**cisco** | **interoperable** | **any**} **Example:** `Device(gdoi-local-server)# rekey acknowledgment interoperable`	Enables group members to acknowledge rekeys. • **cisco**—Accepts Cisco Rekey ACK (encrypted) message.

	Command or Action	Purpose
		• **interoperable**—Requests and accepts interoperable rekey ACK (unencrypted) message. • **any**—Accepts a supported ACK message based on group key member version.
Step 6	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 7	**show crypto gkm ks replay** **Example:** `Device# show crypto gkm ks replay`	Displays rekey acknowledgment configuration on the key server.

Example

The following is a sample output from **show** commands displaying the rekey acknowledgment configuration:

```
Device# show crypto gkm

GROUP INFORMATION
    Group Name               : GETVPN (Unicast)
 .
 .
 .
    Group Rekey Lifetime     : 86400 secs
    Group Rekey
        Remaining Lifetime   : 44710 secs
        Time to Rekey        : 44485 secs
        Acknowledgement Cfg  : {Cisco|Interoperable|Any}
 .
 .
 .

Device# show crypto gkm ks

Total group members registered to this box: 0
Key Server Information For Group GETVPN:
    Group Name               : GETVPN
Group Name               : GETVPN (Unicast)
 .
 .
 .
    Group Members            : 0
      GDOI Group Members     : 0
      G-IKEv2 Group Members  : 0
    Rekey Acknowledgement Cfg: {Cisco|Interoperable|Any}
    IPSec SA Direction       : Both
 .
 .
 .
Device# show crypto gkm ks rekey

Group GETVPN (Unicast)
```

```
    Acknowledgement Type In-Use         : {Cisco|Interoperable|Any}
    Number of Rekeys sent               : 20
 .
 .
 .
Device# show crypto gkm ks rekey

Group GETVPN (Multicast)
    Acknowledgement Type In-Use         : None
    Number of Rekeys sent               : 20
 .
 .
 .
Device# show crypto gkm ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):
  # of teks : 2  Seq num : 7
  KEK POLICY (transport type : Unicast)
    spi : 0x7D32D2052B87CEFE14060B58B0176129
    management alg     : disabled    encrypt alg       : AES
    crypto iv length   : 16          key size          : 16
    orig life(sec): 86400       remaining life(sec): 44699
    time to rekey (sec): 44474
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys
    acknowledgement    : {cisco|interoperable|any}

Device# show crypto gkm ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):
  # of teks : 2  Seq num : 7
  KEK POLICY (transport type : Multicast)
    spi : 0x7D32D2052B87CEFE14060B58B0176129
    management alg     : disabled    encrypt alg       : AES
    crypto iv length   : 16          key size          : 16
    orig life(sec): 86400       remaining life(sec): 44699
    time to rekey (sec): 44474
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys
    acknowledgement    : none
```

Configuration Examples for GET VPN Interoperability

Example: Enabling IP-D3P on a Key Server

```
Device> enable
Device# configure terminal
Device(config)# crypto gkm group GETVPN
Device(config-gkm-group)# server local
Device(gdoi-local-server)# sa d3p window msec 5000
Device(gdoi-local-server)# exit
```

Example: Enabling IP-D3P on a Group Member

```
Device> enable
Device# configure terminal
Device(config-gkm-group)# client d3p window sec 50
Device(gdoi-local-server)# exit
```

Example: Enabling Rekey Acknowledgement

```
Device> enable
Device# configure terminal
Device(config)# crypto gkm group GET
Device(config-gkm-group)# server local
Device(gdoi-local-server)# rekey acknowledgment interoperable
Device(gdoi-local-server)# exit
```

Additional References for GET VPN Interoperability

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
GET VPN configuration	*Cisco Group Encrypted Transport VPN*
Unicast rekey	"Unicast Rekeying" section in the *GET VPN* module

Standards and RFCs

Standard/RFC	Title
draft-weis-delay-detection-00	*IP Delivery Delay Detection Protocol*
draft-weis-gdoi-rekey-ack-01	*GDOI GROUPKEY-PUSH Acknowledgement Message*
RFC 5374- Section 5.4 - Group Associated Policy	*Multicast Extensions to the Security Architecture for the Internet Protocol*
RFC 6407 - Section 4.2.1 - Activation Time Delay	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GET VPN Interoperability

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 313: Feature Information for GET VPN Interoperability

Feature Name	Releases	Feature Information
D3P support on GETVPN Key Server		The D3P support on GETVPN Key Server feature enables support for IP-D3P on a GET VPN network. The following commands were introduced or modified: **client d3p**, **sa d3p**, **show crypto gkm gm replay**, **show crypto gkm ks replay**.
Internet-Draft ACK for Cisco GETVPN Key Server		The Internet-Draft ACK for Cisco GETVPN Key Server implements the standard for rekey acknowledgment message between non-Cisco group members and key server as defined in the GDOI GROUPKEY-PUSH Acknowledgment Message draft. The following commands were introduced or modified: **rekey acknowledgement**, **show crypto gkm**.
RFC 8263 ID Ack implementation		The Group Domain of Interpretation (GDOI) includes the ability of key server to provide a set of current devices with additional security associations. For example, to rekey expiring security associations. This feature adds the ability of a key server to request that the group devices return an acknowledgement of receipt of its rekey message and specifies the acknowledgement method.

CHAPTER 231

Perfect Forward Secrecy for GETVPN

If a Group Member (GM) is compromised, an attacker may access saved long-term keys and messages. With Perfect Forward Secrecy (PFS) for GETVPN, the attacker cannot use the keys and messages to obtain the keys of past or future sessions. Thus, the attacker may use the compromised Traffic Encryption Key (TEK) to decrypt the communication of the current session, but cannot decrypt recorded or future communication.

Feature Information for PFS for GETVPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 314: Feature Information for PFS for GETVPN

Feature Name	Releases	Feature Information
Perfect Forward Secrecy for GETVPN	Cisco IOS XE Gibraltar 16.12.1	The following commands are introduced or modified: **show crypto gkm feature pfs**, **pfs**, **show crypto gdoi**, and **client pfs**.

Information About PFS for GETVPN

Overview of PFS for GETVPN

Suppose a device is compromised and an attacker accesses the long-terms keys saved on it. For Perfect Forward Secrecy (PFS), the attacker must not be able to use the long-term keys to obtain keys and decrypt recorded communication of past sessions. A related security measure is called Perfect Backward Secrecy (PBS). For PBS, the attacker must not be able to use the long-term keys to obtain keys and decrypt communication of future sessions.

If a GM is compromised, an attacker may access saved long-term keys and messages. The attacker may so obtain the Diffie Hellman (DH) result and the registration message or the Key Encryption Key (KEK) and past rekey messages. With PFS for GETVPN, the attacker cannot use the keys and messages to obtain TEKs of past sessions. In addition, the attacker cannot use the KEK to decrypt future rekey messages, and so, cannot obtain TEKs of future sessions. Thus, the attacker has access only to the TEK of the current session. Despite the compromised keys, any recorded or future communication remains secure.

PFS for GETVPN comprises the following changes:

- A modified rekey process; the GM registration mechanism is unchanged.
- If you enable PFS for GETVPN, the default lifetime of the GM-KS IKEv2 channel changes from 1 day to 600 seconds.

  However, if you have configured customized lifetime, the lifetime does not change after you enable PFS for GETVPN.
- Key Servers (KSs) and Group Members (GMs) that support PFS for GETVPN have new version numbers. The version numbers support backward compatibility and interactions with third-party GMs.

Restrictions for PFS for GETVPN

- A Key Server (KS) and a Group Member (GM) must communicate using the IKEv2 protocol. PFS for GETVPN is not supported when KS and GM communicate using the IKEv1 protocol.
- Enable PFS on all the KSs in a COOP. On a GM, PFS is enabled by default.
- Both scheduled rekey and manually triggered rekey cause a GM to reregister with a KS. The reregistration may cause a noticeable overhead at the KS, especially at scale.
- Force rekey can cause traffic loss because of a key mismatch between GMs.
- When the RSA key size is 4096, because of the large size of the key, the Crypto Engine takes considerable time for a rekey signing. During a rekey signing, if too many registration requests are received, the Crypto Engine may be overloaded. An overloaded Crypto Engine logs the following error message:

  ```
  %ACE-3-TRANSERR: IOSXE-ESG(9): IKEA trans 0x11A8; opcode 0x23; param 0x0; error 0xC;
  retry cnt 0
  ```

  You may see this error message more frequently in a GETVPN deployment that uses an RSA key size of 4096, has more than 100 GMs, and has PFS enabled. The increased frequency is because when PFS is enabled, every rekey triggers a re-registration, and with more than a 100 GMs, the Crypto Engine is more likely to receive several registration requests during a rekey signing and may be overloaded.

  Similarly, you may see this error message more frequently in such a deployment if you run the **crypto gdoi ks rekey replace-now** command repeatedly due to the registration requests triggered by this command.

  In a GETVPN deployment with PFS enabled, we recommend that you use an RSA key size of 2048. Using an RSA key size of 4096 is not necessary because the rekey message does not contain TEK/KEK keys.

Modified Rekey Process

PFS for GETVPN ensures that an attacker cannot use the KEK from a compromised GM to decrypt past or future rekey messages. Thus, the attacker cannot obtain past or future KEKs or TEKs. For this purpose, PFS for GETVPN modifies the rekey mechanism so that rekey messages do not include KEKs or TEKs. The contents of the rekey message depend on the type of rekey.

Scheduled Rekey

1. When the rekey timer for KEK or TEK expires, KS generates a new KEK or TEK, respectively.
2. KS sets a private attribute in the GSA payload of the rekey message and encrypts the message with the current KEK. The rekey message does not include the new KEK or TEK. KS sends the rekey message to the GM.
3. GM receives the rekey message and decrypts the message using the current KEK. GM identifies the scheduled rekey and starts a reregistration timer for a random time interval in the 0–6 seconds range.
4. When the reregistration timer expires, GM initiates reregistration with the KS.
5. After reregistration, KS sends the KEK or TEK to the GM over the IKEv2 channel.
6. On receiving a new KEK, GM replaces the old KEK with the new KEK.
7. On receiving a new TEK, GM checks the Activation Time Delay (ATD) for the TEK. If the ATD is non-zero, GM starts a timer to enforce ATD before installing the TEK in the data plane.

   ATD is calculated on the KS as follows:

   a. If Long SA lifetime is configured, the ATD timer is initialized to a value in seconds computed as follows:
   ATD = (remaining lifetime of old TEK) - (1% of remaining lifetime of old TEK) - 75

      The new TEK rolls over at (1% of remaining lifetime of old TEK).

   b. If Long SA lifetime is not configured, the ATD timer is initialized to a value is seconds computed as follows:
   ATD = (remaining lifetime of old TEK) - 75

      The new TEK rolls over 30 seconds before the expiry of the old TEK.

Sync-up Rekey

1. KS sends a GM a rekey message that includes only the pseudoTimeStamp (PST) value.

   The message does not contain KEK or TEK.
2. On receiving the rekey message, GM updates its pseudotime value and does not trigger any reregistration. Depending on the pseudoTimeStamp value received from the KS and the TimeBased Anti Replay (TBAR) window configured on the GM, GM may generate syslog messages.

Manually Triggered Key

1. When you trigger a rekey operation using **crypto gdoi ks** or **clear crypto gdoi ks members**, KS sends the GAP/DELETE payload based on the rekey type.

   • Rekey message without policy change

Table 315: GAP/DELETE Payload for a Rekey Message Without Policy Change

Type	KEK	TEK	Private Attribute for Rekey	KD	GAP	DELETE
`crypto gdoi ks rekey`	No	No	No	No	No	No
`crypto gdoi ks rekey replace-now`	No	No	Yes	No	ATD 1 sec	No
`clear crypto gdoi ks members`	No	No	No	No	ATD 5% of TEK	Yes
`clear crypto gdoi ks members now`	No	No	Yes	No	ATD 1 sec	Yes

   • Rekey message with policy change

Table 316: GAP/DELETE Payload for a Rekey Message with Policy Change

Type	KEK	TEK	Private Attribute for Rekey	KD	GAP	DELETE
`crypto gdoi ks rekey`	No	No	Yes	No	ATD 5% of TEK	No
`crypto gdoi ks rekey replace-now`	No	No	Yes	No	ATD 1 sec	No
`clear crypto gdoi ks members`	No	No	No	No	ATD 5% of TEK	Yes
`clear crypto gdoi ks members now`	No	No	Yes	No	ATD 1 sec	Yes

2. On receiving the rekey message, GM initiates reregistration with the KS.
3. As part of GM reregistration, KS sends the KEK or TEK to the GM over the IKEv2 channel.
4. GM sets the lifetime of the old key to the Activation Time Delay (ATD) value sent by the KS. After the ATD, GM deletes the old key and installs the new key.

Suite-B Support

During the first registration of a GM, KS assigns a unique Sender Identifier (SID) and Initialization Vector (IV) range to the GM. When a GM re-registers with a KS in response to a rekey message, the GM provides the SID that the KS assigned during registration. KS does not assign new a SID or Initialization Vector (IV) range to the GM.

KS and GM Versions for PFS for GETVPN

If Cisco IOS XE Gibraltar 16.12.1 or a later release is installed on a GM, PFS for GETVPN is enabled by default. You can disable PFS for GETVPN using the command-line interface. The GM version varies based on whether PFS is enabled or not as summarized in the following table.

	Without Suite-B Support	With Suite-B Support	ASR 1000 Series
PFS disabled	16	17	19
PFS enabled	21	22	20

If Cisco IOS XE Gibraltar 16.12.1 or a later release is installed on a KS, PFS for GETVPN is disabled by default and the KS version is 1.0.18. You can enable PFS for GETVPN via the CLI. With PFS for GETVPN enabled, the KS version is 1.0.23. Enable PFS for GETVPN on all the cooperative KSs.

KS sends rekey messages to GMs based on the GM version:

- To GMs that have PFS for GETVPN disabled and send a version number such as 1.0.17 or 1.0.19, KS sends rekey messages that have the KEK or TEK.

  KS sends rekey messages that have the KEK or TEK to GMs that have PFS for GETVPN disabled and to non-Cisco GMs. GMs that have PFS for GETVPN disabled send a version number such as 1.0.17 or 1.0.19 to the KS. Non-Cisco GMs send an unknown version number to the KS.

- KS sends modified rekey messages that do not have KEK or TEK to GMs that have PFS for GETVPN enabled. GMs that have PFS for GETVPN enabled send a version number such as 1.0.20 or 1.0.22

Upgrading KS and GM for PFS for GETVPN

For PFS for GETVPN to be effective, enable PFS on every KS and GM in the network. If you do not enable PFS for GETVPN on a GM and the GM is breached, compromised keys can hamper the security of the entire network.

Upgrade the KSs in the network as follows:

Note We recommend that you upgrade KSs while there is sufficient time for the expiry of KEK and TEKs.

1. Upgrade a Secondary KS and wait for the COOP election to complete.
2. Repeat Step 1 for each Secondary KS in the COOP.

   The Secondary KSs reboot and synchronize with the Primary KS and assume the role of Secondary KSs.

3. Upgrade the Primary KS.

   One of the Secondary KSs is elected as the new Primary KS. The upgraded KS reboots and assumes the role of a Secondary KS.

4. Enable PFS on all the KSs.

After the upgrade, KS sends rekey messages based on the version number that GMs send. Based on the GM version number, KS sends rekey messages that contain KEK or TEK, or modified rekey messages without KEK or TEK.

INDEX

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V