JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Common Criteria Guide for SRX5400, SRX5600, and SRX5800 Devices

Published
2022-10-07

RELEASE
22.2R1

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Common Criteria Guide for SRX5400, SRX5600, and SRX5800 Devices
22.2R1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | viii

1 Overview

Understanding the Common Criteria Evaluated Configuration | 2

Understanding Junos OS in FIPS Mode of Operation | 4

Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms | 6

Identifying Secure Product Delivery | 9

Applying Tamper-Evident Seals to the Cryptographic Module | 10

Understanding Management Interfaces | 13

2 Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS in FIPS Mode of Operation | 16

Understanding Services for Junos OS in FIPS Mode of Operation | 18

Downloading Software Packages from Juniper Networks | 23

Installing Junos Software Packages | 23

Understanding Zeroization to Clear System Data for FIPS Mode of Operation | 24

Loading Firmware on the Device | 26

How to Enable and Configure Junos OS in FIPS Mode of Operation | 26

3 Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | 32

Configuring a Network Device Protection Profile Authorized Administrator | 34

4 Network Time Protocol

NTP Overview | 37

Network Time Security (NTS) Support for NTP | 38

NTP Time Servers | 41

Configure NTP Time Server and Time Services | 42

- Configure the Router or Switch to Operate in Client Mode | 43
- Configure the Router or Switch to Operate in Symmetric Active Mode | 44
- Configure the Router or Switch to Operate in Broadcast Mode | 44
- Configure the Router or Switch to Operate in Server Mode | 45

Example: Configure NTP as a Single Time Source for Router and Switch Clock Synchronization | 46

Synchronize and Coordinate Time Distribution Using NTP | 47

- Configure NTP | 47
- Configure NTP Boot Server | 48
- Specify a Source Address for an NTP Server | 49

NTP Configuration | 51

Example: Configure NTP | 54

- Requirements | 54
- Overview | 55
- Configuration | 55
- Verification | 57

NTP Authentication Keys | 59

Configure Devices to Listen to Broadcast Messages Using NTP | 60

Configure Devices to Listen to Multicast Messages Using NTP | 60

5 Configuring SSH and Console Connection

Understanding FIPS Authentication Methods | 63

Configuring a System Login Message and Announcement | 64

Limiting the Number of User Login Attempts for SSH Sessions | 65

Configuring SSH on the Evaluated Configuration | 66

6 Configuring the Remote Syslog Server

Sample Syslog Server Configuration on a Linux System | 70

Configuring Event Logging to a Local File | 70

Configuring Event Logging to a Remote Server | 71

Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server | 71

Forwarding Logs to the External Syslog Server | 77

7 Configuring Audit Log Options

Configuring Audit Log Options in the Evaluated Configuration | 79

Configuring Audit Log Options for SRX5400, SRX5600, and SRX5800 Devices | 79

Sample Code Audits of Configuration Changes | 80

8 Configuring Event Logging

Event Logging Overview | 85

Interpreting Event Messages | 90

Logging Changes to Secret Data | 91

Login and Logout Events Using SSH | 93

Logging of Audit Startup | 94

9 Configuring a Secure Logging Channel

Creating a Secure Logging Channel | 96

10 Configuring VPNs

Configuring VPN on a Device Running Junos OS | 104

11 Configuring Security Flow Policies

Understanding a Security Flow Policy on a Device Running Junos OS | 128

12 Configuring Traffic Filtering Rules

Overview | 133

Understanding Protocol Support | 133

Configuring Traffic Filter Rules | 135

Configuring Default Deny-All and Reject Rules | 136

Logging the Dropped Packets Using Default Deny-all Option | 137

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets | 138

Configuring Default Reject Rules for Source Address Spoofing | 139

Configuring Default Reject Rules with IP Options | 140

Configuring Default Reject Rules | 141

13 Configuring Network Attacks

Configuring IP Teardrop Attack Screen | 143

Configuring TCP Land Attack Screen | 144

Configuring ICMP Fragment Screen | 146

Configuring Ping-Of-Death Attack Screen | 148

Configuring tcp-no-flag Attack Screen | 150

Configuring TCP SYN-FIN Attack Screen | 151

Configuring TCP fin-no-ack Attack Screen | 153

Configuring UDP Bomb Attack Screen | 155

Configuring UDP CHARGEN DoS Attack Screen | 155

Configuring TCP SYN and RST Attack Screen | 157

Configuring ICMP Flood Attack Screen | 159

Configuring TCP SYN Flood Attack Screen | 161

Configuring TCP Port Scan Attack Screen | 163

Configuring UDP Port Scan Attack Screen | 165

Configuring IP Sweep Attack Screen | 167

14 Configuring the IDP Extended Package

IDP Extended Package Configuration Overview | 170

15 Configuring Cluster Mode

Understanding Cluster Mode | 172

Configuring L2 HA Link Encryption tunnel | 172

Configuring PKI Based L2HA Link Encryption | 179

16 Performing Self-Tests on a Device

Understanding FIPS Self-Tests | 191

17 Configuration Statements

checksum-validate | 204

code | 206

data-length | 207

destination-option | 209

extension-header | 211

header-type | 212

home-address | 214

identification | 216

icmpv6 (Security IDP Custom Attack) | 218

ihl (Security IDP Custom Attack) | 220

option-type | 221

reserved (Security IDP Custom Attack) | 223

routing-header | 225

sequence-number (Security IDP ICMPv6 Headers) | 226

type (Security IDP ICMPv6 Headers) | 228

About This Guide

Use this guide to configure and evaluate SRX5400, SRX5600, and SRX5800 for Common Criteria (CC) compliance. Common Criteria for information technology is an international agreement signed by several countries that permit the evaluation of security products against a common set of standards.

RELATED DOCUMENTATION

- Common Criteria and FIPS Certifications

CHAPTER 1

Overview

Understanding the Common Criteria Evaluated Configuration | 2

Understanding Junos OS in FIPS Mode of Operation | 4

Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms | 6

Identifying Secure Product Delivery | 9

Applying Tamper-Evident Seals to the Cryptographic Module | 10

Understanding Management Interfaces | 13

Understanding the Common Criteria Evaluated Configuration

IN THIS SECTION

- Understanding Common Criteria | 3
- Supported Platforms | 3

This document describes the steps required to duplicate the configuration of the device running Junos OS when the device is evaluated. This is referred to as the evaluated configuration. The following list describes the standards to which the device has been evaluated:

- Collaborative Protection Profile for Network Devices, NDcPPv2.2e—https://www.commoncriteriaportal.org/files/ppfiles/CPP_ND_V2.2E.pdf.

  PP modules for NDcPP are as follows:

  - MOD_FW_CPP v1.4e –https://www.niap-ccevs.org/MMO/PP/MOD_CPP_FW_v1.4e.pdf
  - MOD_IPS_V1.0 –https://www.niap-ccevs.org/MMO/PP/MOD_IPS_v1.0.pdf
  - VPNGW_MOD v1.1 – https://www.niap-ccevs.org/MMO/PP/mod_vpngw_v1.1.pdf

- Network Device Collaborative Protection Profile (NDcPPv2.2)/Stateful Traffic Filter Firewall Collaborative Protection Profile (FWcPP) Extended Package VPN Gateway, Version 2.2, 22 March 2020 (VPNEP)
- Collaborative Protection Profile for Stateful Traffic Filter Firewalls, version 2.0, 14 March 2018 (FWcPP)https://www.commoncriteriaportal.org/files/ppfiles/CPP_FW_V2.0E.pdf
- Collaborative Protection Profile for Network Devices or Collaborative Protection Profile for Stateful Traffic Filter Firewalls Extended Package (EP) for Intrusion Prevention Systems (IPS), (IPSEP)
- FIPS—https://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

These documents are available at https://www.niap-ccevs.org/Profile/PP.cfm.

NOTE: On SRX5400, SRX5600, and SRX5800 devices, Junos OS Release 22.2R1 is certified for Common Criteria with FIPS mode enabled on the devices.

Understanding Common Criteria

Common Criteria for information technology is an international agreement signed by several countries that permits the evaluation of security products against a common set of standards. In the Common Criteria Recognition Arrangement (CCRA) at http://www.commoncriteriaportal.org/ccra/, the participants agree to mutually recognize evaluations of products performed in other countries. All evaluations are performed using a common methodology for information technology security evaluation.

For more information on Common Criteria, see http://www.commoncriteriaportal.org/.

Supported Platforms

For the features described in this document, the following platforms are supported:

- The IPSEP, NDcPP, FWcPP, and VPNEP apply to:
  - SRX5400, SRX5600, and SRX5800

Understanding Junos OS in FIPS Mode of Operation

IN THIS SECTION

- About the Cryptographic Boundary on Your Device | **5**
- How FIPS Mode of Operation Differs from Non-FIPS Mode of Operation | **5**
- Validated Version of Junos OS in FIPS Mode of Operation | **6**

Federal Information Processing Standards (FIPS) 140-3 defines security levels for hardware and software that perform cryptographic functions. Junos-FIPS is a version of the Junos operating system (Junos OS) that complies with Federal Information Processing Standard (FIPS) 140-3.

Operating SRX Series devices in a FIPS 140-3 Level 2 environment requires enabling and configuring FIPS mode of operation on the device from the Junos OS command-line interface (CLI).

The *Security Administrator* enables FIPS mode of operation in Junos OS Release 22.2R1 and sets up keys and passwords for the system and other *FIPS users* who can view the configuration. Both user types can also perform normal configuration tasks on the device (such as modify interface types) as individual user configuration allows.

BEST PRACTICE: Be sure to verify the secure delivery of your device and apply tamper-evident seals to its vulnerable ports.

About the Cryptographic Boundary on Your Device

FIPS 140-3 compliance requires a defined *cryptographic boundary* around each *cryptographic module* on a device. Junos OS in FIPS mode of operation prevents the cryptographic module from running any software that is not part of the FIPS-certified distribution, and allows only FIPS-approved cryptographic algorithms to be used. No critical security parameters (CSPs), such as passwords and keys, can cross the cryptographic boundary of the module by, for example, being displayed on a console or written to an external log file.

CAUTION: Virtual Chassis features are not supported in FIPS mode of operation. Do not configure a Virtual Chassis in FIPS mode of operation.

To physically secure the cryptographic module, all Juniper Networks devices require a tamper-evident seal on the USB and mini-USB ports.

How FIPS Mode of Operation Differs from Non-FIPS Mode of Operation

Unlike Junos OS in non-FIPS mode of operation, Junos OS in FIPS mode of operation is a *nonmodifiable operational environment*. In addition, Junos OS in FIPS mode of operation differs in the following ways from Junos OS in non-FIPS mode of operation:

- Self-tests of all cryptographic algorithms are performed at startup.
- Self-tests of random number and key generation are performed continuously.
- Weak cryptographic algorithms such as Data Encryption Standard (DES) and MD5 are disabled.
- Weak, remote, or unencrypted management connections must not be configured.
- Passwords must be encrypted with strong one-way algorithms that do not permit decryption.
- Junos-FIPS administrator passwords must be at least 10 characters long.
- Cryptographic keys must be encrypted before transmission.

The FIPS 140-3 standard is available for download from the National Institute of Standards and Technology (NIST) at http://csrc.nist.gov/publications/fips/fips140-3/fips1402.pdf.

Validated Version of Junos OS in FIPS Mode of Operation

To determine the validated version of Junos OS in FIPS mode of operation and for regulatory compliance information about Common Criteria and FIPS for Juniper Networks products, see the Juniper Networks Compliance Advisor.

Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms

IN THIS SECTION

- FIPS Terminology | 6
- Supported Cryptographic Algorithms | 8

Use the definitions of FIPS terms and supported algorithms to help you understand Junos OS in FIPS mode of operation.

FIPS Terminology

Critical security parameter (CSP)	Security-related information—for example, secret and private cryptographic keys and authentication data such as passwords and personal identification numbers (PINs)—whose disclosure or modification can compromise the security of a cryptographic module or the information it protects.
Cryptographic module	The set of hardware, software, and firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained

Security Administrator	within the cryptographic boundary. SRX Series devices are certified at FIPS 140-3 Level 2.
ESP	Person with appropriate permissions who is responsible for securely enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode of operation on a device. For details, see Understanding Roles and Services for Junos OS in Common Criteria and FIPS Mode.
FIPS	Encapsulating Security Payload (ESP) protocol. The part of the IPsec protocol that guarantees the confidentiality of packets through encryption. The protocol ensures that if an ESP packet is successfully decrypted, and no other party knows the secret key the peers share, the packet was not wiretapped in transit.
IKE	Federal Information Processing Standards. FIPS 140-3 specifies requirements for security and cryptographic modules. Junos OS in FIPS mode of operation complies with FIPS 140-3 Level 2.
IPsec	The Internet Key Exchange (IKE) is part of IPsec and provides ways to securely negotiate the shared private keys that the authentication header (AH) and ESP portions of IPsec need to function properly. IKE employs Diffie-Hellman key-exchange methods and is optional in IPsec. (The shared keys can be entered manually at the endpoints.)
KATs	The IP Security (IPsec) protocol. A standard way to add security to Internet communications. An IPsec security association (SA) establishes secure communication with another FIPS cryptographic module by means of mutual authentication and encryption.
SA	Known answer tests. System self-tests that validate the output of cryptographic algorithms approved for FIPS and test the integrity of some Junos OS modules. For details, see Understanding FIPS Self-Tests.
SPI	Security association (SA). A connection between hosts that allows them to communicate securely by defining, for example, how they exchange private keys. As Security Administrator, you must manually configure an internal SA on devices running Junos OS in FIPS mode of operation. All values, including the keys, must be statically specified in the configuration.
	Security parameter index (SPI). A numeric identifier used with the destination address and security protocol in IPsec to identify an SA. Because you manually configure the SA for Junos OS in FIPS mode of operation, the SPI must be entered as a parameter rather than derived randomly.

SSH A protocol that uses strong authentication and encryption for remote access across a nonsecure network. SSH provides remote login, remote program execution, file copy, and other functions. It is intended as a secure replacement for `rlogin`, `rsh`, and `rcp` in a UNIX environment. To secure the information sent over administrative connections, use SSHv2 for CLI configuration. In Junos OS, SSHv2 is enabled by default, and SSHv1, which is not considered secure, is disabled.

Zeroization Erasure of all CSPs and other user-created data on a device before its operation as a FIPS cryptographic module—or in preparation for repurposing the device for non-FIPS operation. The Security Administrator can zeroize the system with a CLI operational command. For details, see "Understanding Zeroization to Clear System Data for FIPS Mode of Operation" on page 24.

Supported Cryptographic Algorithms

Each implementation of an algorithm is checked by a series of known answer test (KAT) self-tests. Any self-test failure results in a FIPS error state.

> **BEST PRACTICE**: For FIPS 140-3 compliance, use only FIPS-approved cryptographic algorithms in Junos OS in FIPS mode of operation.

The following cryptographic algorithms are supported in FIPS mode of operation. Symmetric methods use the same key for encryption and decryption, while asymmetric methods (preferred) use different keys for encryption and decryption.

AES The Advanced Encryption Standard (AES), defined in FIPS PUB 197. The AES algorithm uses keys of 128, 192, or 256 bits to encrypt and decrypt data in blocks of 128 bits.

Diffie-Hellman A method of key exchange across a nonsecure environment (such as the Internet). The Diffie-Hellman algorithm negotiates a session key without sending the key itself across the network by allowing each party to pick a partial key independently and send part of that key to the other. Each side then calculates a common key value. This is a symmetrical method, and keys are typically used only for a short time, discarded, and regenerated.

ECDH Elliptic Curve Diffie-Hellman. A variant of the Diffie-Hellman key exchange algorithm that uses cryptography based on the algebraic structure of elliptic curves over finite fields. ECDH allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. The shared secret can be used either as a key or to derive another key for encrypting subsequent communications using a symmetric key cipher.

ECDSA Elliptic Curve Digital Signature Algorithm. A variant of the Digital Signature Algorithm (DSA) that uses cryptography based on the algebraic structure of elliptic curves over finite fields. The bit size of the elliptic curve determines the difficulty of decrypting the key. The public key believed to be needed for ECDSA is about twice the size of the security level, in bits. ECDSA using the P-256, P-384, or the P-521 curve can be configured under OpenSSH.

HMAC Defined as "Keyed-Hashing for Message Authentication" in RFC 2104, HMAC combines hashing algorithms with cryptographic keys for message authentication. For Junos OS in FIPS mode of operation, HMAC uses the iterated cryptographic hash function SHA-1 (designated as HMAC-SHA1) along with a secret key.

RELATED DOCUMENTATION

Understanding Zeroization to Clear System Data for FIPS Mode of Operation | 24

Understanding FIPS Self-Tests

Identifying Secure Product Delivery

There are several mechanisms provided in the delivery process to ensure that a customer receives a product that has not been tampered with. The customer should perform the following checks upon receipt of a device to verify the integrity of the platform.

- Shipping label—Ensure that the shipping label correctly identifies the correct customer name and address as well as the device.
- Outside packaging—Inspect the outside shipping box and tape. Ensure that the shipping tape has not been cut or otherwise compromised. Ensure that the box has not been cut or damaged to allow access to the device.
- Inside packaging—Inspect the plastic bag and seal. Ensure that the bag is not cut or removed. Ensure that the seal remains intact.

If the customer identifies a problem during the inspection, he or she should immediately contact the supplier. Provide the order number, tracking number, and a description of the identified problem to the supplier.

Additionally, there are several checks that can be performed to ensure that the customer has received a box sent by Juniper Networks and not a different company masquerading as Juniper Networks. The customer should perform the following checks upon receipt of a device to verify the authenticity of the device:

- Verify that the device was ordered using a purchase order. Juniper Networks devices are never shipped without a purchase order.
- When a device is shipped, a shipment notification is sent to the e-mail address provided by the customer when the order is taken. Verify that this e-mail notification was received. Verify that the e-mail contains the following information:
  - Purchase order number
  - Juniper Networks order number used to track the shipment
  - Carrier tracking number used to track the shipment
  - List of items shipped including serial numbers
  - Address and contacts of both the supplier and the customer
- Verify that the shipment was initiated by Juniper Networks. To verify that a shipment was initiated by Juniper Networks, you should perform the following tasks:
  - Compare the carrier tracking number of the Juniper Networks order number listed in the Juniper Networks shipping notification with the tracking number on the package received.
  - Log on to the Juniper Networks online customer support portal at https://support.juniper.net/support/ to view the order status. Compare the carrier tracking number or the Juniper Networks order number listed in the Juniper Networks shipment notification with the tracking number on the package received.

RELATED DOCUMENTATION

Understanding the Common Criteria Evaluated Configuration | 2

Applying Tamper-Evident Seals to the Cryptographic Module

IN THIS SECTION

General Tamper-Evident Seal Instructions | 11

SRX5400 Device Tamper-Evident Seal Application | 12
SRX5600 Device Tamper-Evident Seal Application | 12
SRX5800 Device Tamper-Evident Seal Application | 13

The cryptographic modules physical embodiment is that of a multi-chip standalone device that meets Level 2 physical security requirements. The module is completely enclosed in a rectangular nickel or clear zinc coated, cold rolled steel, plated steel, and brushed aluminum enclosure. There are no ventilation holes, gaps, slits, cracks, slots, or crevices that would allow for any sort of observation of any component contained within the cryptographic boundary. Tamper-evident seals allow the operator to verify if the enclosure has been breached. These seals are not factory-installed and must be applied by the Cryptographic Officer.

NOTE: Seals are available for order from Juniper Networks using part number JNPR-FIPS-TAMPER-LBLS.

As a Cryptographic Officer, you are responsible for:

- Applying seals to secure the cryptographic module
- Controlling any unused seals
- Controlling and observing any changes, such as repairs or booting from an external USB drive to the cryptographic module, that require removing or replacing the seals to maintain the security of the module

As per the security inspection guidelines, upon receipt of the cryptographic module, the Cryptographic Officer must check that the labels are free of any tamper evidence.

General Tamper-Evident Seal Instructions

All FIPS certified devices require a tamper-evident seal on the USB ports. While applying seals, follow these general instructions:

- Handle the seals with care. Do not touch the adhesive side. Do not cut or otherwise resize a seal to make it fit.
- Make sure all surfaces to which the seals are applied are clean and dry and clear of any residue.

- Apply the seals with firm pressure across the seal to ensure adhesion. Allow at least 1 hour for the adhesive to cure.

The following sections describe the tamper-evident seal application method for SRX5400, SRX5600, and SRX5800 devices.

SRX5400 Device Tamper-Evident Seal Application

On SRX5400 devices, apply 13 tamper-evident seals at the following locations:

Front Pane:

- Apply two seals vertically, connecting them to the topmost (non-honeycomb) subpane. Position the seals so that they extend to the thin pane below and the honeycomb panel above.
- Apply one seal vertically across the thin pane, extending to the blank pane below and the subpane above.
- Apply three seals vertically, one on each long horizontal subpane. Position each seal so that it attaches to the subpane above and the one below (or to the chassis, if it is bottommost subpane). Ensure that one of the seals extends to the left subpane below the thin subpane.

Back Pane:

- Apply four seals vertically, one on each of the top four subpanes, extending to the large chassis plate below.
- Apply one seal vertically on the horizontal screwed-in plate that rests on the large central chassis. Position the seal so that it extends to the chassis in both directions.
- Apply two seals horizontally on the low side of the subpanes. Position the seals so that they extend to the large central chassis area and wrap around to the neighboring side panes.

SRX5600 Device Tamper-Evident Seal Application

On SRX5600 devices, apply 18 tamper-evident seals at the following locations:

Front Pane:

Apply 11 seals vertically, one seal on each horizontal subpane (excluding the honeycomb plate on the top and the thin subpane below), second seal on the top (non-honeycomb) subpane, and one more seal at the bottom. Position the seals so that they attach to vertically adjacent subpanes. The seals should

attach to the vertically adjacent subpanes. Position the bottom seal so that it attaches to the lowermost subpane and wraps around, attaching to the bottom pane. Ensure that one of the seals spans across the thin plate with ample extra distance on each side.

Back Pane:

- Apply five seals vertically, one on each of the top four subpanes, attaching to the large plate below.
- Apply two seals horizontally, one on each of the vertical side subpanes, extending to both the large central plate and the side panes.

SRX5800 Device Tamper-Evident Seal Application

On SRX5800 devices, apply 24 tamper-evident seals at the following locations:

Front Pane:

- Apply fourteen seals horizontally, one on each of the long vertical subpanes, extending to the neighboring two subpanes. If on an end subpane, seal should wrap around to the side.
- Apply three seal vertically, one seal covering each of the thin panes, two seals near the bottom, and one seal near the top of the lower half.
- Apply two seals vertically, both on the console area at the top of the module, one seal extending to the top and the other seal extending to the chassis area below.

Back Pane:

Apply five seals horizontally, three seals spanning the gaps between the vertical subpanes, and then two more seals, one seal on each of the far edges of the left and right panels. The last two seals must wrap around to the sides.

RELATED DOCUMENTATION

How to Enable and Configure Junos OS in FIPS Mode of Operation

Understanding Management Interfaces

The following management interfaces can be used in the evaluated configuration:

- Local Management Interfaces—The RJ-45 console port on the rear panel of a device is configured as RS-232 data terminal equipment (DTE). You can use the command-line interface (CLI) over this port to configure the device from a terminal.
- Remote Management Protocols—The device can be remotely managed over any Ethernet interface. SSHv2 is the only permitted remote management protocol that can be used in the evaluated configuration, and it is enabled by default on the device. The remote management protocols J-Web and Telnet are not available for use on the device in the evaluated configuration.

RELATED DOCUMENTATION

Understanding the Common Criteria Evaluated Configuration | 2

2

CHAPTER

Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS in FIPS Mode of Operation | 16

Understanding Services for Junos OS in FIPS Mode of Operation | 18

Downloading Software Packages from Juniper Networks | 23

Installing Junos Software Packages | 23

Understanding Zeroization to Clear System Data for FIPS Mode of Operation | 24

Loading Firmware on the Device | 26

How to Enable and Configure Junos OS in FIPS Mode of Operation | 26

Understanding Roles and Services for Junos OS in FIPS Mode of Operation

IN THIS SECTION

- Security Administrator Role and Responsibilities | 16
- FIPS User Role and Responsibilities | 17
- What Is Expected of All FIPS Users | 18

The Juniper Networks Junos operating system (Junos OS) running in non-FIPS mode of operation allows a wide range of capabilities for users, and authentication is identity-based. In contrast, the FIPS 140-3 standard defines two user roles: *Security Administrator* and *FIPS user*. These roles are defined in terms of Junos OS user capabilities.

All other user types defined for Junos OS in FIPS mode of operation (operator, administrative user, and so on) must fall into one of the two categories: Security Administrator or FIPS user. For this reason, user authentication in FIPS mode of operation is role-based rather than identity-based.

In addition to their FIPS roles, both user types can perform normal configuration tasks on the device as individual user configuration allows.

Security Administrators and FIPS users perform all FIPS-related configuration tasks and issue all statements and commands for Junos OS in FIPS mode of operation. Security Administrator and FIPS user configurations must follow the guidelines for Junos OS in FIPS mode of operation.

For details, see:

Security Administrator Role and Responsibilities

The Security Administrator is the person responsible for enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode of operation on a device. The Security Administrator securely installs Junos OS on the device, enables FIPS mode of operation, establishes keys and passwords for other users and software modules, and initializes the device before network connection. The Security Administrator can configure and monitor the module through a console or SSH connection.

BEST PRACTICE: We recommend that the Security Administrator administer the system in a secure manner by keeping passwords secure and checking audit files.

The permissions that distinguish the Security Administrator from other FIPS users are `secret`, `security`, `maintenance`, and `control`. For FIPS compliance, assign the Security Administrator to a login class that contains all of these permissions. A user with the Junos OS maintenance permission can read files containing critical security parameters (CSPs).

NOTE: Junos OS in FIPS mode of operation does not support the *FIPS 140-3 maintenance role*, which is different from the Junos OS maintenance permission.

Among the tasks related to Junos OS in FIPS mode of operation, the Security Administrator is expected to:

- Set the initial root password.
- Reset user passwords for FIPS-approved algorithms during upgrades from Junos OS.
- Set up manual IPsec SAs for configuration with dual Routing Engines.
- Examine log and audit files for events of interest.
- Erase user-generated files and data on (zeroize) the device.

FIPS User Role and Responsibilities

All FIPS users, including the Security Administrator, can view the configuration. Only the user assigned as the Security Administrator can modify the configuration.

The permissions that distinguish Security Administrators from other FIPS users are `secret`, `security`, `maintenance`, and `control`. For FIPS compliance, assign the FIPS user to a class that contains *none* of these permissions.

FIPS users configure networking features on the device and perform other tasks that are not specific to FIPS mode of operation. FIPS users who are not Security Administrators can perform reboots and view status output.

What Is Expected of All FIPS Users

All FIPS users, including the Security Administrator, must observe security guidelines at all times.

All FIPS users must:

- Keep all passwords confidential.
- Store devices and documentation in a secure area.
- Deploy devices in secure areas.
- Check audit files periodically.
- Conform to all other FIPS 140-3 security rules.
- Follow these guidelines:
  - Users are trusted.
  - Users abide by all security guidelines.
  - Users do not deliberately compromise security.
  - Users behave responsibly at all times.

RELATED DOCUMENTATION

Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms | 6

Understanding Zeroization to Clear System Data for FIPS Mode of Operation | 24

Understanding Services for Junos OS in FIPS Mode of Operation

IN THIS SECTION

Understanding Authenticated Services | 19

Critical Security Parameters | 20

All services implemented by the module are listed in the tables that follow.

Understanding Authenticated Services

Table 1 on page 19 lists the authenticated services on the device running Junos OS.

Table 1: Authenticated services

Authenticated Services	Description	Security Administrator	User (read-only)	User (network)
Configure security	Security relevant configuration	x	–	–
Configure	Non-security relevant configuration	x	–	–
Secure traffic	IPsec protected routing	–	–	x
Status	Display the status	x	x	–
Zeroize	Destroy all critical security parameters (CSPs)	x	–	–
SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	x	x	–
IPsec connect	Initiate IPsec connection (IKE)	x	–	x
Console access	Console monitoring and control (CLI)	x	x	–

Table 1: Authenticated services *(Continued)*

Authenticated Services	Description	Security Administrator	User (read-only)	User (network)
Remote reset	Software-initiated reset	x	–	–

Table 2: Unauthenticated traffic

Service	Description
Local reset	Hardware reset or power cycle
Traffic	Traffic requiring no cryptographic services

Critical Security Parameters

Critical security parameters (CSPs) are security-related information such as cryptographic keys and passwords that can compromise the security of the cryptographic module or the security of the information protected by the module if they are disclosed or modified.

Zeroization of the system erases all traces of CSPs in preparation for operating the device as a cryptographic module.

Table 3 on page 20 lists the CSP access rights within services.

Table 3: CSP Access Rights Within Services

Service	CSPs					
	DRBG_Seed	DRBG_State	SSH PHK	SSH DH	SSH-SEK	ESP-SEK
Configure security	–	E	G, W	–	–	–

Table 3: CSP Access Rights Within Services *(Continued)*

Service	CSPs					
	DRBG_Seed	DRBG_State	SSH PHK	SSH DH	SSH-SEK	ESP-SEK
Configure	–	–	–	–	–	–
Secure Traffic	–	–	–	–	–	E
Status	–	–	–	–	–	–
Zeroize	Z	Z	Z	Z	Z	Z
SSH connect	–	E	E	G, E	G, E	–
IPSec connect	–	E	–	–	–	G
Console access	–	–	–	–	–	–
Remote reset	G, E	G	–	Z	Z	Z
Local Reset	G, E	G	–	Z	Z	Z
Traffic	–	–	–	–	–	–

Service	CSPs				
	IKE-PSK	IKE-Priv	IKE-SKEYI	IKE-SKE	IKE-DH-PRI
Configure security	W	G, W	–	–	–
Configure	–	–	–	–	–

(Continued)

Service	CSPs				
	IKE-PSK	IKE-Priv	IKE-SKEYI	IKE-SKE	IKE-DH-PRI
Secure Traffic	–	–	–	E	–
Status	–	–	–	–	–
Zeroize	Z	Z	–	–	–
SSH connect	–	–	–	–	–
IPSec connect	E	E	G	G	G
Console access	–	–	–	–	–
Remote reset	–	–	Z	Z	Z
Local Reset	–	–	Z	Z	Z
Traffic	–	–	–	–	–

Here:

- G = Generate: The device generates the CSP.
- E = Execute: The device runs using the CSP.
- W = Write: The CSP is updated or written to the device.
- Z = Zeroize: The device zeroizes the CSP.

RELATED DOCUMENTATION

Understanding Zeroization to Clear System Data for FIPS Mode of Operation | 24

Understanding FIPS Authentication Methods | 63

Downloading Software Packages from Juniper Networks

To operate in Junos OS in FIPS mode, the device must have the following software package installed. You can download the following Junos OS software packages from the Juniper Networks website:

- Junos OS for SRX5400, SRX5600, and SRX5800 Release 22.2R1.

Before you begin to download the software, ensure that you have a Juniper Networks Web account and a valid support contract. To obtain an account, complete the registration form at the Juniper Networks website:

https://userregistration.juniper.net/entitlement/setupAccountInfo.do.

To download software packages from Juniper Networks:

1. Using a Web browser, follow the links to the download URL on the Juniper Networks webpage. https://support.juniper.net/support/downloads/
2. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by Juniper Networks representatives.
3. Download the software. See Downloading Software

RELATED DOCUMENTATION

Installation and Upgrade Guide

Installing Junos Software Packages

SRX Series devices can provide the security defined by Federal Information Processing Standards (FIPS) 140-3 Level 2 if these devices are operated in the Junos OS in FIPS mode.

NOTE: Junos OS is delivered in signed packages that contain digital signatures to ensure the Juniper Networks software is running. When installing the software packages, Junos OS validates the signatures and the public key certificates used to digitally sign the software packages. If the signature or certificate is found to be invalid (for example, when the certificate validity period has

expired or cannot be verified against the root CA stored in the Junos OS internal store), the installation process fails.

To install these software packages, perform the following tasks:

1. Download the Junos OS package and the Junos FIPS mode package from https://support.juniper.net/support/downloads/. See Downloading Software.
2. Install the Junos OS on your device using a TFTP server, see Installing Junos OS on SRX Series Devices from the Boot Loader Using a TFTP Server or install Junos OS on your device using the following CLI command: `request system software add /<image-path>/<junos package> no-copy no-validate reboot`.

RELATED DOCUMENTATION

- Installation and Upgrade Guide

Understanding Zeroization to Clear System Data for FIPS Mode of Operation

IN THIS SECTION

- Why Zeroize? | 25
- When to Zeroize? | 25

Zeroization completely erases all configuration information on the device, including all plaintext passwords, secrets, and private keys for SSH, local encryption, local authentication, and IPsec. To exit the FIPS mode you need to zeroize the device.

The cryptographic module provides a non-approved mode of operation in which non-approved cryptographic algorithms are supported. When moving from the non-approved mode of operation to the approved mode of operation, the Security Administrator must zeroize the non-approved mode critical security parameters (CSPs). For SRX5400, SRX5600, and SRX5800 devices, the Security Administrator

initiates the zeroization process by entering the `request vmhost zeroize` command from the CLI after enabling FIPS mode of operation. Use of this command is restricted to the Security Administrator.

> ⚠ **CAUTION**: Perform system zeroization with care. After the zeroization process is complete, no data is left on the device. This command erases all the CSPs, configurations, and the hard disk partitions containing the device image. Hence, the device does not boot up on zeroization and USB reimage is required to recover the device.

Zeroization can be time-consuming. Although all configurations are removed in a few seconds, the zeroization process goes on to overwrite all media, which can take considerable time depending on the size of the media.

Why Zeroize?

Your device is not considered a valid FIPS cryptographic module until all CSPs have been entered—or reentered—while the device is in FIPS mode of operation. For FIPS 140-3 compliance, the only way to exit from FIPS mode is to zeroize the TOE.

When to Zeroize?

As a Security Administrator, perform zeroization in the following situations:

- **Before FIPS operation**—To prepare your device for operation as a FIPS cryptographic module, perform zeroization to remove the non-approved mode critical security parameters (CSPs) and enable FIPS mode on the device.
- **Before non-FIPS operation**—To begin repurposing your device for non-FIPS operation, perform zeroization on the device.

  > **NOTE**: Juniper Networks does not support installing non-FIPS software in a FIPS mode of operation, but doing so might be necessary in certain test environments. Be sure to zeroize the system first.

- **When a tamper-evident seal is disturbed**—If the seal on an insecure port has been tampered with, the system is considered to be compromised. After applying new tamper-evident seals to the appropriate locations, zeroize the system and set up new passwords and CSPs.

Loading Firmware on the Device

The Junos OS 22.2R1 FIPS images only accept the firmware signed with ECDSA and rejects any firmware signed with RSA+SHA1. You cannot downgrade to images that are signed with RSA+SHA1 from "ECDSA signed only" images. In this scenario, the SRX Series device does not load the firmware.

RELATED DOCUMENTATION

How to Enable and Configure Junos OS in FIPS Mode of Operation

How to Enable and Configure Junos OS in FIPS Mode of Operation

To enable the Junos OS in FIPS mode of operation, perform the following steps:

1. Zeroize the device before enabling FIPS mode of operation

   `user@host>` **request vmhost zeroize**

2. Enable the FIPS mode on the device.

   `user@host#` **set system fips level 2**

3. Set the root password.

   `user@host#` **set system root-authentication plain-text-password.**

   Enter a password.

4. Remove the CSPs on commit check.

   `user@host#` **commit**

5. After you reboot the device, perform integrity and self-test when the module is operating in FIPS mode.

6. Configure IKEv2 when AES-GCM is used for encryption of IKE and/or IPSec.

```
user@host# set security ike proposal <ike_proposal_name> encryption-algorithm ?
Possible completions:
aes-128-cbc AES-CBC 128-bit encryption algorithm
aes-128-gcm AES-GCM 128-bit encryption algorithm
aes-192-cbc AES-CBC 192-bit encryption algorithm
aes-256-cbc AES-CBC 256-bit encryption algorithm
aes-256-gcm AES-GCM 256-bit encryption algorithm
user@host# set security ike proposal <ike_proposal_name> encryption-algorithm aes-256-gcm
user@host# set security ipsec proposal <ipsec_proposal_name> encryption-algorithm aes-128-gcm
user@host# set security ike gateway <gateway_name> version ?
Possible completions:
v1-only              The connection must be initiated using IKE version 1
v2-only              The connection must be initiated using IKE version 2
user@host# set security ike gateway <gateway_name> version v2-only
user@host# commit
commit complete
```

Ensure that the backup image of the firmware is also a JUNOS-FIPS image by issuing the `request system snapshot` command.

NOTE:

`show configuration security ikeshow configuration security ipsec`

```
user@host-srx5400:fips> show version
      Hostname: host-srx5400
      Model: srx5400
      Junos: 22.2R1.9
      JUNOS OS Kernel 64-bit [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS libs [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS runtime [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS time zone information [20220607.2c547a1_builder_stable_12_222]
      JUNOS network stack and utilities [20220617.153850_builder_junos_222_r1]
      JUNOS libs [20220617.153850_builder_junos_222_r1]
      JUNOS OS libs compat32 [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS 32-bit compatibility [20220607.2c547a1_builder_stable_12_222]
      JUNOS libs compat32 [20220617.153850_builder_junos_222_r1]
      JUNOS runtime [20220617.153850_builder_junos_222_r1]
      Junos vmguest package [20220617.153850_builder_junos_222_r1]
```

```
      JUNOS py extensions [20220617.153850_builder_junos_222_r1]
      JUNOS py base [20220617.153850_builder_junos_222_r1]
      JUNOS OS vmguest [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS crypto [20220607.2c547a1_builder_stable_12_222]
      JUNOS OS boot-ve files [20220607.2c547a1_builder_stable_12_222]
      JUNOS na telemetry [22.2R1.9]
      JUNOS Web Management Platform Package [20220617.153850_builder_junos_222_r1]
      JUNOS srx libs compat32 [20220617.153850_builder_junos_222_r1]
      JUNOS srx runtime [20220617.153850_builder_junos_222_r1]
      JUNOS Routing mpls-oam-basic [20220617.153850_builder_junos_222_r1]
      JUNOS Routing lsys [20220617.153850_builder_junos_222_r1]
      JUNOS Routing 32-bit Compatible Version [20220617.153850_builder_junos_222_r1]
      JUNOS Routing aggregated [20220617.153850_builder_junos_222_r1]
      Redis [20220617.153850_builder_junos_222_r1]
      JUNOS probe utility [20220617.153850_builder_junos_222_r1]
      JUNOS common platform support [20220617.153850_builder_junos_222_r1]
      JUNOS srx platform support [20220617.153850_builder_junos_222_r1]
      JUNOS Openconfig [22.2R1.9]
      JUNOS mtx network modules [20220617.153850_builder_junos_222_r1]
      JUNOS modules [20220617.153850_builder_junos_222_r1]
      JUNOS srx modules [20220617.153850_builder_junos_222_r1]
      JUNOS srx libs [20220617.153850_builder_junos_222_r1]
      JUNOS L2 RSI Scripts [20220617.153850_builder_junos_222_r1]
      JUNOS srx Data Plane Crypto Support [20220617.153850_builder_junos_222_r1]
      JUNOS ike [20220617.153850_builder_junos_222_r1]
      JUNOS daemons [20220617.153850_builder_junos_222_r1]
      JUNOS srx daemons [20220617.153850_builder_junos_222_r1]
      JUNOS High End AppQos Daemon [20220617.153850_builder_junos_222_r1]
      JUNOS Services URL Filter package [20220617.153850_builder_junos_222_r1]
      JUNOS Services TLB Service PIC package [20220617.153850_builder_junos_222_r1]
      JUNOS Services Telemetry [20220617.153850_builder_junos_222_r1]
      JUNOS Services TCP-LOG [20220617.153850_builder_junos_222_r1]
      JUNOS Services SSL [20220617.153850_builder_junos_222_r1]
      JUNOS Services SOFTWIRE [20220617.153850_builder_junos_222_r1]
      JUNOS Services Stateful Firewall [20220617.153850_builder_junos_222_r1]
      JUNOS Services RTCOM [20220617.153850_builder_junos_222_r1]
      JUNOS Services RPM [20220617.153850_builder_junos_222_r1]
      JUNOS Services PCEF package [20220617.153850_builder_junos_222_r1]
      JUNOS Services NAT [20220617.153850_builder_junos_222_r1]
      JUNOS Services Mobile Subscriber Service Container package
[20220617.153850_builder_junos_222_r1]
      JUNOS Services MobileNext Software package [20220617.153850_builder_junos_222_r1]
      JUNOS Services Logging Report Framework package [20220617.153850_builder_junos_222_r1]
```

```
      JUNOS Services LL-PDF Container package [20220617.153850_builder_junos_222_r1]
      JUNOS Services Jflow Container package [20220617.153850_builder_junos_222_r1]
      JUNOS Services Deep Packet Inspection package [20220617.153850_builder_junos_222_r1]
      JUNOS Services IPSec [20220617.153850_builder_junos_222_r1]
      JUNOS Services IDS [20220617.153850_builder_junos_222_r1]
      JUNOS IDP Services [20220617.153850_builder_junos_222_r1]
      JUNOS Services HTTP Content Management package [20220617.153850_builder_junos_222_r1]
      JUNOS Services DNS Filter package (i386) [20220617.153850_builder_junos_222_r1]
      JUNOS Services Crypto [20220617.153850_builder_junos_222_r1]
      JUNOS Services Captive Portal and Content Delivery Container package
[20220617.153850_builder_junos_222_r1]
      JUNOS Services COS [20220617.153850_builder_junos_222_r1]
      JUNOS AppId Services [20220617.153850_builder_junos_222_r1]
      JUNOS Services Application Level Gateways [20220617.153850_builder_junos_222_r1]
      JUNOS Services AACL Container package [20220617.153850_builder_junos_222_r1]
      JUNOS Extension Toolkit [20220617.153850_builder_junos_222_r1]
      JUNOS Packet Forwarding Engine Support (wrlinuxlts19)
[20220617.153850_builder_junos_222_r1]
      JUNOS Packet Forwarding Engine Support (spc3) [20220617.153850_builder_junos_222_r1]
      JUNOS Packet Forwarding Engine Support (MX/EX92XX Common)
[20220617.153850_builder_junos_222_r1]
      JUNOS Packet Forwarding Engine Support (M/T Common) [20220617.153850_builder_junos_222_r1]
      JUNOS Packet Forwarding Engine Support (MX Common) [20220617.153850_builder_junos_222_r1]
      JUNOS Juniper Malware Removal Tool (JMRT) [1.0.0+20220617.153850_builder_junos_222_r1]
      JUNOS J-Insight [20220617.153850_builder_junos_222_r1]
      JUNOS jfirmware [20220608.110139_builder_junos_222_r1]
      JUNOS Online Documentation [20220617.153850_builder_junos_222_r1]
      JUNOS jail runtime [20220607.2c547a1_builder_stable_12_222]
      JUNOS fips optest [22.2R1.9]
      JUNOS FIPS mode utilities [20220617.153850_builder_junos_222_r1]
      JUNOS dsa dsa [22.2R1.9]
```

The fips keyword next to the `hostname` in the output indicates that the module is operating in FIPS mode for Junos Software Release 22.2R1.

```
      user@host-vSRX3.0:fips> show configuration security ike
      proposal ike-proposal1 {
      authentication-method pre-shared-keys;
      dh-group group14;
      encryption-algorithm aes-256-gcm;
```

```
        }
        policy ike-policy1 {
        mode main;
        proposals ike-proposal1;
        pre-shared-key ascii-text "$9$Hq.5zF/tpBUj9Au0IRdbwsaZ"; ## SECRET-DATA
        }
        gateway gw1 {
        ike-policy ike-policy1;
        address 198.51.100.0;
        local-identity inet 203.0.113.0;
        external-interface ge-0/0/3;
        version v2-only;
        }
        user@host-vSRX3.0:fips> show configuration security ipsec
        proposal ipsec-proposal1 {
        protocol esp;
        encryption-algorithm aes-128-gcm;
        }
        policy ipsec-policy1 {
        perfect-forward-secrecy {
        keys group14;
        }
        proposals ipsec-proposal1;
        }
        vpn vpn1 {
        bind-interface st0.0;
        ike {
        gateway gw1;
        ipsec-policy ipsec-policy1;
```

RELATED DOCUMENTATION

Loading Firmware on the Device | 26

CHAPTER 3

Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | 32

Configuring a Network Device Protection Profile Authorized Administrator | 34

Understanding the Associated Password Rules for an Authorized Administrator

The authorized administrator is associated with a defined login class, and the administrator is assigned with all permissions. Data is stored locally for fixed password authentication.

NOTE: We recommend that you not use control characters in passwords.

Use the following guidelines and configuration options for passwords and when selecting passwords for authorized administrator accounts. Passwords should be:

- Easy to remember so that users are not tempted to write it down.
- Changed periodically.
- Private and not shared with anyone.
- Contain a minimum of 10 characters. The minimum password length is 10 characters.

```
[ edit ]
administrator@host# set system login password minimum-length 10
```

- Include both alphanumeric and punctuation characters, composed of any combination of upper and lowercase letters, numbers, and special characters such as, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. There should be at least a change in one case, one or more digits, and one or more punctuation marks.
- Contain character sets. Valid character sets include uppercase letters, lowercase letters, numbers, punctuation, and other special characters.

```
[ edit ]
administrator@host# set system login password change-type character-sets
```

- Contain the minimum number of character sets or character set changes. The minimum number of character sets required in plain-text passwords in Junos FIPS is 2.

```
[ edit ]
administrator@host# set system login password minimum-changes 2
```

NOTE: The authentication algorithm for plain-text passwords must be configured as sha256.

```
[ edit ]
administrator@host# set system login password format sha256
```

Weak passwords are:

- Words that might be found in or exist as a permuted form in a system file such as **/etc/passwd**.
- The hostname of the system (always a first guess).
- Any words appearing in a dictionary. This includes dictionaries other than English, and words found in works such as Shakespeare, Lewis Carroll, Roget's Thesaurus, and so on. This prohibition includes common words and phrases from sports, sayings, movies, and television shows.
- Permutations on any of the above. For example, a dictionary word with vowels replaced with digits (for example f00t) or with digits added to the end.
- Any machine-generated passwords. Algorithms reduce the search space of password-guessing programs and so should not be used.

Strong reusable passwords can be based on letters from a favorite phrase or word, and then concatenated with other, unrelated words, along with additional digits and punctuation.

NOTE: Passwords should be changed periodically.

RELATED DOCUMENTATION

Understanding Junos OS in FIPS Mode of Operation | 4

Identifying Secure Product Delivery | 9

Configuring a Network Device Protection Profile Authorized Administrator

An account for `root` is always present in a configuration and is not intended for use in normal operation. In the evaluated configuration, the `root` account is restricted to the initial installation and configuration of the evaluated device.

An NDPP authorized administrator must have all permissions, including the ability to change the router configuration.

To configure an authorized administrator:

1. Create a login class named security-admin with all permissions.

```
[edit]
root@host# set system login class security-admin permissions all
```

2. Define your NDPP user authorized administrator.

```
[edit]
root@host# set system login user NDcPPv2-user class security-admin authentication encrypted-password
```

OR

```
root@host# set system login user NDcPPv2-user class security-admin authentication plain-text-password
```

3. Configure the authentication algorithm for plain-text passwords as sha256.

```
[edit]
root@host# set system login password format sha256
```

4. Commit the changes.

```
[edit]
root@host# commit
```

NOTE: The root password should be reset following the change to sha256 for the password storage format. This ensures the new password is protected using a sha256 hash, rather than the default password hashing algorithm. To reset the root password, use the `set system login user root password` *password* command, and confirm the new password when prompted.

RELATED DOCUMENTATION

Understanding the Associated Password Rules for an Authorized Administrator | 32

4

CHAPTER

Network Time Protocol

NTP Overview | 37

NTP Time Servers | 41

Configure NTP Time Server and Time Services | 42

Example: Configure NTP as a Single Time Source for Router and Switch Clock Synchronization | 46

Synchronize and Coordinate Time Distribution Using NTP | 47

NTP Configuration | 51

Example: Configure NTP | 54

NTP Authentication Keys | 59

Configure Devices to Listen to Broadcast Messages Using NTP | 60

Configure Devices to Listen to Multicast Messages Using NTP | 60

NTP Overview

IN THIS SECTION

- Network Time Security (NTS) Support for NTP | 38

Network Time Protocol (NTP) is a widely used protocol used to synchronize the clocks of routers and other hardware devices on the Internet. Primary NTP servers are synchronized to a reference clock directly traceable to Coordinated Universal Time (UTC). Reference clocks include GPS receivers and telephone modem services, NTP accuracy expectations depend on the environment application requirements. However, NTP can generally maintain time to within tens of milliseconds over the public internet.

NTP is defined in the RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification

Devices running Junos OS can be configured to act as an NTP client, a secondary NTP server, or a primary NTP server. These variations are as follows:

- Primary NTP Server—Primary NTP servers are synchronized to a reference clock that is directly traceable to UTC. These servers then re-distribute this time data downstream to other Secondary NTP servers or NTP clients.
- Secondary NTP Server—Secondary NTP servers are synchronized to a primary or secondary NTP server. These servers then re-distribute this data downstream to other Secondary NTP servers or NTP clients.
- NTP Client—NTP clients are synchronized to a primary or secondary NTP server. Clients do not re-distribute this time data to other devices.

NOTE: The NTP subnet includes a number of widely accessible public primary time servers that can be used as a network's primary NTP server. Juniper Networks strongly recommends that you authenticate any primary servers you use.

Each device on your network can be configured to run in one or more of the following NTP modes:

- Broadcast Mode—One or more devices is set up to transmit time information to a specified broadcast or multicast address. Other devices listen for time sync packets on these addresses. This mode is less accurate than the client/server mode.

- Client/Server Mode—Devices are organized hierarchically across the network in client/server relationships.
- Symmetric Active (peer) Mode—Two or more devices are configured as NTP server peers to provide redundancy.

By default, if an NTP client time drifts so that the difference in time from the NTP server exceeds 128 milliseconds, the NTP client is automatically stepped back into synchronization. The NTP client will still synchronize with the server even if the offset between the NTP client and server exceeds the 1000-second threshold. You can manually request that a device synchronize with an NTP server by using the `set date ntp` operational command on the router. On devices running Junos OS that have dual Routing Engines, the backup Routing Engine synchronizes directly with the primary Routing Engine.

All Juniper platforms that run Junos OS support the leap second adjustment. By default, if the NTP server is aware of the leap second calculations, then the Junos device will automatically add the 1 second delay. PTP (Precision Time Protocol) is used to detect and propagate leap second synchronization changes throughout all nodes in a network. NTP is also required for Common Criteria compliance. For more information on the Common Criteria certification, see Public Sector Certifications.

For more details about the Network Time Protocol, go to the Network Time Foundation website at http://www.ntp.org.

NTP supports IPv4 VPN and IPv6 routing and forwarding (VRF) requests on Junos OS. VRF request is also supported on Junos OS Evolved Release 20.2R1 onwards. This enables an NTP server running on a provider edge (PE) router to respond to NTP requests from a customer edge (CE) router. As a result, a PE router can process any NTP request packet coming from different routing instances.

Network Time Security (NTS) Support for NTP

IN THIS SECTION

- NTS Overview | **39**
- Benefits of NTS | **39**
- Network Time Synchronization with NTS | **39**

NTS Overview

NTS provides cryptographic security for network time synchronization and supports client-server mode of NTP. NTS uses Transport Layer Security (TLS) protocol and Authenticated Encryption with Associated Data (AEAD) to obtain network time in an authenticated manner to the users. NTS also provides support for encryption of NTP extension fields.

The most important security processes are dependent on accurate time. Network time synchronization from a malicious source leads to serious consequences. Enabling NTS ensures accurate network time synchronization on your device.

Benefits of NTS

- Provides strong cryptographic protection against wide range of security attacks such as packet manipulation, spoofing, DDOS amplification attacks, and replay attacks
- Ensures accurate network time synchronization from a reliable source
- Provides scalability: Servers can serve several clients without manually pre-configuring any client-specific configuration. Because of the usage of cookies the server does not need to locally store the client specific data such as keys and AEAD algorithm
- Prevents tracking of mobile devices

Network Time Synchronization with NTS

NTS consists of two protocols, the NTS Key Establishment protocol (NTS-KE) and the NTP time synchronization using NTS extension fields.

NTS-KE Protocol

In the NTS-KE protocol phase, the NTS-KE protocol manages the initial authentication, NTS parameter negotiation, and key establishment over TLS in the following order:

1. The client performs a TLS handshake with the NTS-KE server and successfully verify the certificates.
2. The client performs the NTS parameters negotiation with the server over the TLS-protected channel. The cryptographic algorithms negotiated are AEAD methods, which protects the NTP packets in the second phase.
3. The client and the server successfully establish the key material for communication.
4. The server also sends a supply of initial cookies to the client to use in next phase.

5. The TLS channel closes and NTP proceeds to the next phase where actual exchange of time data happens.

NTS supports only the TLS version 1.3. The older TLS versions get rejected during the NTS-KE protocol phase.

NTP Time Synchronization Using NTS Extension Fields

This phase manages the encryption and authentication during NTP time synchronization through the extension fields in the NTP packets in the following order:

1. The client queries the NTP server about time with NTS extension fields. These extension fields include cookies and an authentication tag computed using negotiated AEAD algorithm and key material extracted from the NTS-KE handshake.

   An NTS-secured NTP client request contains the following NTS extension fields:

   - **Unique Identifier Extension Field**: Contains randomly generated data and provides the means for replay protection at the NTS level.
   - **NTS Cookie Extension Field**: Contains the information about the key material, which establishes during NTS-KE phase, and the negotiated cryptographic algorithm. A cookie is only used once in a request to prevent tracking.
   - **NTS Cookie Placeholder Extension Field**: (Optional) Communicates to the server that the client wants to receive additional cookies in the response packet.
   - **NTS Authenticator and Encrypted Extension Fields**: Generated using AEAD Algorithm and key established during NTS-KE. This field provides the integrity protection for the NTP header and all the previous extension fields.

   Constant refreshing of cookies protects a device from tracking when it changes network addresses. For example a mobile device moving across different networks. The lack of any recognizable data prevents an adversary from determining that two packets sent over different network addresses came from the same client.

2. When the server receives an NTS-secured request from the client, the server decrypts the cookie with a master key.

3. The server extracts the negotiated AEAD algorithm and the keys that are available in the cookie. Using this key, the server checks the integrity of the NTP packet to ensure no manipulations to the packet.

4. The server generates one or more new cookies and creates the NTP response packet. The server generates at least one new cookie and one additional cookie for each Cookie Placeholder Extension Field that the client added in the request packet.

The response packet contains two NTS extension fields:

- The Unique Identifier Extension Field, which has the same contents from the Unique Identifier field in request packet.
- The NTS Authenticator and Encrypted Extension Field, which secures the NTP header and the previous extension fields using the extracted keys.

5. The server also encrypts the cookies and includes them in the NTS Authenticator and Encrypted Extension Fields. This procedure also protects the client from tracking because an attacker cannot extract the cookies from a response message.
6. The server finalizes the response packet and sends the packet to the client.
7. The client receives the response packet.
8. The client checks the Unique Identifier field and verifies that the Unique Identifier matches with an outstanding request.
9. The client successfully performs the integrity check of the packet using the key and the AEAD algorithm.
10. The client decrypts the cookies and adds them to its pool and processes the time information received from server.

NTP Time Servers

The IETF defined the Network Time Protocol (NTP) to synchronize the clocks of computer systems connected to each other over a network. Most large networks have an NTP server that ensures that time on all devices is synchronized, regardless of the device location. If you use one or more NTP servers on your network, ensure you include the NTS server addresses in your Junos OS configuration.

When configuring the NTP, you can specify which system on the network is the authoritative time source, or time server, and how time is synchronized between systems on the network. To do this, you configure the router, switch, or security device to operate in one of the following modes:

- Client mode—In this mode, the local router or switch can be synchronized with the remote system, but the remote system can never be synchronized with the local router or switch.
- Symmetric active mode—In this mode, the local router or switch and the remote system can synchronize with each other. You use this mode in a network in which either the local router or switch or the remote system might be a better source of time.

Symmetric active mode can be initiated by either the local or the remote system. Only one system needs to be configured to do so. This means that the local system can synchronize with any system that offers symmetric active mode without any configuration whatsoever. However, we strongly encourage you to configure authentication to ensure that the local system synchronizes only with known time servers.

- Broadcast mode—In this mode, the local router or switch sends periodic broadcast messages to a client population at the specified broadcast or multicast address. Normally, you include this statement only when the local router or switch is operating as a transmitter.
- Server mode—In this mode, the local router or switch operates as an NTP server.

  In NTP server mode, the Junos OS supports authentication as follows:

  - If the NTP request from the client comes with an authentication key (such as a key ID and message digest sent with the packet), the request is processed and answered based on the authentication key match.
  - If the NTP request from the client comes without any authentication key, the request is processed and answered without authentication.

Configure NTP Time Server and Time Services

IN THIS SECTION

- Configure the Router or Switch to Operate in Client Mode | 43
- Configure the Router or Switch to Operate in Symmetric Active Mode | 44
- Configure the Router or Switch to Operate in Broadcast Mode | 44
- Configure the Router or Switch to Operate in Server Mode | 45

When you use NTP, configure the router or switch to operate in one of the following modes:

- Client mode
- Symmetric active mode
- Broadcast mode
- Server mode

The following topics describe how to configure these modes of operation:

Configure the Router or Switch to Operate in Client Mode

To configure the local router or switch to operate in client mode, include the `server` statement and other optional statements at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
server address <key key-number> <version value> <prefer>;
authentication-key key-number type type value password;
trusted-key[key-numbers];
```

Specify the address of the system acting as the time server. You must specify an address, not a hostname.

To include an authentication key in all messages sent to the time server, include the **key** option. The key corresponds to the key number you specify in the `authentication-key` statement, as described in .

By default, the router or switch sends NTP version 4 packets to the time server. To set the NTP version level to 1, 2, or 3, include the **version** option.

If you configure more than one time server, you can mark one server preferred by including the **prefer** option.

The following example shows how to configure the router or switch to operate in client mode:

```
[edit system ntp]
authentication-key 1 type md5 value "$ABC123";
server 10.1.1.1 key 1 prefer;
trusted-key 1;
```

Configure the Router or Switch to Operate in Symmetric Active Mode

To configure the local router or switch to operate in symmetric active mode, include the `peer` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
peer address <key key-number> <version value> <prefer>;
```

Specify the address of the remote system. You must specify an address, not a hostname.

To include an authentication key in all messages sent to the remote system, include the **key** option. The key corresponds to the key number you specify in the `authentication-key` statement.

By default, the router or switch sends NTP version 4 packets to the remote system. To set the NTP version level to 1, 2 or 3, include the **version** option.

If you configure more than one remote system, you can mark one system preferred by including the **prefer** option:

```
peer address <key key-number> <version value> prefer;
```

Configure the Router or Switch to Operate in Broadcast Mode

To configure the local router or switch to operate in broadcast mode, include the `broadcast` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
broadcast address <key key-number> <version value> <ttl value>;
```

Specify the broadcast address on one of the local networks or a multicast address assigned to NTP. You must specify an address, not a hostname. If the multicast address is used, it must be 224.0.1.1.

To include an authentication key in all messages sent to the remote system, include the key option. The key corresponds to the key number you specify in the `authentication-key` statement.

By default, the router or switch sends NTP version 4 packets to the remote system. To set the NTP version level to 1, 2, or 3, include the version option.

Configure the Router or Switch to Operate in Server Mode

In server mode, the router or switch acts as an NTP server for clients when the clients are configured appropriately. The only prerequisite for " server mode" is that the router or switch must be receiving time from another NTP peer or server. No other configuration is necessary on the router or switch.

When configuring the NTP service in the management VRF (`mgmt_junos`), you must configure at least one IP address on a physical or logical interface within the default routing instance and ensure that this interface is up in order for the NTP service to work with the mgmt_junos VRF.

To configure the local router or switch to operate as an NTP server, include the following statements at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
authentication-key  key-number type type value password;
server address <key key-number> <version value> <prefer>;
trusted-key [key-numbers];
```

Specify the address of the system acting as the time server. You must specify an address, not a hostname.

To include an authentication key in all messages sent to the time server, include the **key** option. The key corresponds to the key number you specify in the `authentication-key` statement.

By default, the router or switch sends NTP version 4 packets to the time server. To set the NTP version level to 1,or 2, or 3, include the **version** option.

If you configure more than one time server, you can mark one server preferred by including the **prefer** option.

The following example shows how to configure the router or switch to operate in server mode:

```
[edit system ntp]
authentication-key 1 type md5 value "$ABC123";
server 192.168.27.46 prefer;
trusted-key 1;
```

Example: Configure NTP as a Single Time Source for Router and Switch Clock Synchronization

Debugging and troubleshooting are much easier when the timestamps in the log files of all the routers or switches are synchronized, because events that span the network can be correlated with synchronous entries in multiple logs. We strongly recommend using the Network Time Protocol (NTP) to synchronize the system clocks of routers, switches, and other network equipment.

By default, NTP operates in an entirely unauthenticated manner. If a malicious attempt to influence the accuracy of a router or switch's clock succeeds, it could have negative effects on system logging, make troubleshooting and intrusion detection more difficult, and impede other management functions.

The following sample configuration synchronizes all the routers or switches in the network to a single time source. We recommend using authentication to make sure that the NTP peer is trusted. The `boot-server` statement identifies the server from which the initial time of day and date is obtained when the router boots. The `server` statement identifies the NTP server used for periodic time synchronization. The `authentication-key` statement specifies that an HMAC-Message Digest 5 (MD5) scheme should be used to hash the key value for authentication, which prevents the router or switch from synchronizing with an attacker's host posing as the time server.

```
[edit]
system {
    ntp {
        authentication-key 2 type md5 value "$ABC123"; # SECRET-DATA
        boot-server 10.1.4.1;
        server 10.1.4.2 key 2;
        trusted key 2;
    }
}
```

Synchronize and Coordinate Time Distribution Using NTP

IN THIS SECTION

- Configure NTP | 47
- Configure NTP Boot Server | 48
- Specify a Source Address for an NTP Server | 49

Using NTP to synchronize and coordinate time distribution in a large network involves these tasks:

Configure NTP

- To configure NTP on the switch, include the `ntp` statement at the `[edit system]` hierarchy level:

```
[edit system]
ntp {
    authentication-key number type type value password;
    boot-server (address | hostname);
    broadcast <address> <key key-number> <version value> <ttl value>;
    broadcast-client;
    multicast-client <address>;
    peer address <key key-number> <version value> <prefer>;
    server address <key key-number> <version value> <prefer>;
    ntp source-address;
    trusted-key [ key-numbers ];
}
```

Configure NTP Boot Server

When you boot the switch, it issues an **ntpdate** request, which polls a network server to determine the local date and time. You need to configure a server that the switch uses to determine the time when the switch boots. Otherwise, NTP will not be able to synchronize to a time server if the server's time appears to be very far off of the local switch's time.

- To configure the NTP boot server, include the `boot-server` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
boot-server (address | hostname);
```

NOTE: The `boot-server` option is deprecated starting in Junos OS Release 20.4R1.

- Junos OS Release 15.1 onwards, to configure the NTP boot server, include the `set ntp server` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
set server (address | hostname);
```

Specify either the IP address or the hostname of the network server.

Specify a Source Address for an NTP Server

For IP version 4 (IPv4), you can specify that if the NTP server configured at the `[edit system ntp]` hierarchy level is contacted on one of the loopback interface addresses, the reply always uses a specific source address. This is useful for controlling which source address NTP will use to access your network when it is either responding to an NTP client request from your network or when it itself is sending NTP requests to your network.

To configure the specific source address that the reply will always use, and the source address that requests initiated by NTP server will use, include the `source-address` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
source-address source-address;
```

source-address is a valid IP address configured on one of the router or switch interfaces.

When configuring the NTP service in the management VRF (`mgmt_junos`), you must configure at least one IP address on a physical or logical interface within the default routing instance and ensure that this interface is up in order for the NTP service to work with the mgmt_junos VRF.

Starting in Junos OS Release 13.3, and Junos OS Evolved Release 20.2R1 you can configure the source address using the `routing-instance` statement at the `[edit system ntp source-address` *`source-address`*`]` hierarchy level:

```
[edit system ntp source-address source-address]
user@host# set routing-instance routing-instance-name
```

For example, the following statement is configured:

```
[edit system ntp source-address source-address]
user@host# set system ntp source-address 12.12.12.12 routing-instance ntp-source-test
```

As a result, while sending NTP message through any interface in the *ntp-source-test* routing instance, the source address 12.12.12.12 is used.

NOTE: The `routing-instance` statement is optional and if not configured, the primary address of the interface will be used.

NOTE: If a firewall filter is applied on the loopback interface, ensure that the `source-address` specified for the NTP server at the `[edit system ntp]` hierarchy level is explicitly included as one of the match criteria in the firewall filter. This enables the Junos OS to accept traffic on the loopback interface from the specified source address.

The following example shows a firewall filter with the source address `10.0.10.100` specified in the `from` statement included at the `[edit firewall filter` *`firewall-filter-name`*`]` hierarchy:

```
[edit firewall filter Loopback-Interface-Firewall-Filter]
term Allow-NTP {
    from {
        source-address {
            172.17.27.46/32; // IP address of the NTP server
            10.0.10.100/32; // Source address specified for the NTP server
            }
        then accept;
    }
}
```

If no source-address is configured for the NTP server, include the primary address of the loopback interface in the firewall filter.

NTP Configuration

The Network Time Protocol (NTP) provides the mechanisms to synchronize time and coordinate time distribution in a large, diverse network. Debugging and troubleshooting are much easier when the timestamps in the log files of all the routers or switches are synchronized, because events that span the network can be correlated with synchronous entries in multiple logs. We recommend using the Network Time Protocol (NTP) to synchronize the system clocks of routers, switches, and other network equipment.

To configure NTP:

1. Configure Junos OS to retrieve the time when it first boots up.

   Use the boot-server statement with the IP address of your NTP server. If DNS is configured, you can use a domain name instead of an IP address.

   ```
   [edit system ntp]
   user@host# set boot-server (name | ip-address)
   ```

   For example, set an IP address of 172.16.1.1 for your NTP server.

   ```
   [edit system ntp]
   user@host# set boot-server 172.16.1.1
   ```

   For example, set a domain name. In this example, the domain name is provided by pool.ntp.org.

   ```
   [edit system ntp]
   user@host# set boot-server 0.north-america.pool.ntp.org
   ```

2. (Optional) Configure one or more reference NTP servers to keep the device synchronized with periodic updates.

It is a good practice to do this, as the Junos OS device can remain up for a long time, and therefore the clock can drift.

```
[edit system ntp]
user@host# set server (name | ip-address)
```

For example, set an IP address of 172.16.1.1 for your NTP server.

```
[edit system ntp]
user@host# set server 172.16.1.1
```

For example, set a domain name provided by pool.ntp.org.

```
[edit system ntp]
user@host# set server 0.north-america.pool.ntp.org
```

3. (Optional) Set the local time zone to match the device's location.

   Universal Coordinated Time (UTC) is the default. Many administrators prefer to keep all their devices configured to use the UTC time zone. This approach has the benefit of allowing you to easily compare the time stamps of logs and other events across a network of devices in many different time zones.

   On the other hand, setting the time zone allows Junos OS to present the time in the correct local format.

   ```
   [edit system ntp]
   user@host# set time-zone time-zone
   ```

   For example:

   ```
   [edit system ntp]
   user@host# set time-zone America/Los_Angeles
   ```

4. Verify the configuration.

Check the system uptime. This command provides the current time, when the device was last booted, when the protocols started, and when the device was last configured.

```
user@host> show system uptime
Current time: 2013-07-25 16:33:38 PDT
System booted: 2013-07-11 17:14:25 PDT (1w6d 23:19 ago)
Protocols started: 2013-07-11 17:16:35 PDT (1w6d 23:17 ago)
Last configured: 2013-07-23 12:32:42 PDT (2d 04:00 ago) by user
 4:33PM  up 13 days, 23:19, 1 user, load averages: 0.00, 0.01, 0.00
```

Check the NTP server status and associations of the clocking sources used by your device.

```
user@host> show ntp associations

     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 tux.brhelwig.co .INIT.          16 -    -  512    0    0.000    0.000 4000.00
```

```
user@host > show ntp status
status=c011 sync_alarm, sync_unspec, 1 event, event_restart,
version="ntpd 4.2.0-a Thu May 30 19:14:15 UTC 2013 (1)",
processor="i386", system="JUNOS13.2-20130530_ib_13_3_psd.1", leap=11,
stratum=16, precision=-18, rootdelay=0.000, rootdispersion=5.130,
peer=0, refid=INIT,
reftime=00000000.00000000  Wed, Feb  6 2036 22:28:16.000, poll=4,
clock=d59d4f2e.1793bce9  Fri, Jul 26 2013 12:40:30.092, state=1,
offset=0.000, frequency=62.303, jitter=0.004, stability=0.000
```

To configure NTP on the router or switch, include the `ntp` statement at the `[edit system]` hierarchy level:

```
[edit system]
ntp {
    authentication-key number type type value password;
    boot-server (address | hostname);
    broadcast <address> <key key-number> <routing-instance-name routing-instance-name> <ttl
value> <version value> ;
    broadcast-client;
    multicast-client <address>;
```

```
    peer address <key key-number> <version value> <prefer>;
    server address <key key-number> <version value> <prefer>;
    source-address <source-address> <routing-instance routing-instance-name>;
    trusted-key [ key-numbers ];
}
```

Example: Configure NTP

IN THIS SECTION

- Requirements | 54
- Overview | 55
- Configuration | 55
- Verification | 57

The Network Time Protocol (NTP) provides the mechanism to synchronize time and coordinate time distribution in a large, diverse network. NTP uses a returnable-time design in which a distributed subnet of time servers operating in a self-organizing, hierarchical primary-secondary configuration synchronizes local clocks within the subnet and to national time standards by means of wire or radio. The servers also can redistribute reference time using local routing algorithms and time daemons.

This example shows how to configure NTP:

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.1 or later
- A switch connected to a network on which an NTP boot server and NTP server reside

Overview

Debugging and troubleshooting are much easier when the timestamps in the log files of all switches are synchronized, because events that span a network can be correlated with synchronous entries in multiple logs. We recommend using the Network Time Protocol (NTP) to synchronize the system clocks of your switch and other network equipment.

In this example, an administrator wants to synchronize the time in a switch to a single time source. We recommend using authentication to make sure that the NTP peer is trusted. The `boot-server` statement identifies the server from which the initial time of day and date are obtained when the switch boots. The `server` statement identifies the NTP server used for periodic time synchronization. The `authentication-key` statement specifies that an HMAC-Message Digest 5 (MD5) scheme is used to hash the key value for authentication, which prevents the switch from synchronizing with an attacker's host that is posing as the time server.

Configuration

IN THIS SECTION

- Procedure | 55

To configure NTP:

Procedure

CLI Quick Configuration

To quickly configure NTP, copy the following commands and paste them into the switch's terminal window:

```
[edit system]
set ntp boot-server 10.1.4.1
set ntp server 10.1.4.2
set ntp authentication-key 2 type md5 value "$ABC123"
```

Step-by-Step Procedure

To configure NTP :

1. Specify the boot server:

```
[edit system]
user@switch# set ntp boot-server 10.1.4.1
```

2. Specify the NTP server:

```
[edit system]
user@switch# set ntp server 10.1.4.2
```

3. Specify the key number, authentication type (MD5), and key for authentication:

```
[edit system]
user@switch# set ntp authentication-key 2 type md5 value "$ABC123"
```

Results

Check the results:

```
[edit system]
user@switch# show
ntp {
    boot-server 10.1.4.1;
    authentication-key 2 type md5 value "$ABC123"; ## SECRET-DATA
    server 10.1.4.2;
}
```

Verification

IN THIS SECTION

- Checking the Time | 57
- Displaying the NTP Peers | 58
- Displaying the NTP Status | 58

To confirm that the configuration is correct, perform these tasks:

Checking the Time

Purpose

Check the time that has been set on the switch.

Action

Enter the `show system uptime` operational mode command to display the time.

```
user@switch> show system uptime
fpc0:
--------------------------------------------------------------------------
Current time: 2009-06-12 12:49:03 PDT
System booted: 2009-05-15 06:24:43 PDT (4w0d 06:24 ago)
Protocols started: 2009-05-15 06:27:08 PDT (4w0d 06:21 ago)
Last configured: 2009-05-27 14:57:03 PDT (2w1d 21:52 ago) by admin1
12:49PM  up 28 days,  6:24, 1 user, load averages: 0.05, 0.06, 0.01
```

Meaning

The output shows that the current date and time are June 12, 2009 and 12:49:03 PDT. The switch booted 4 weeks, 6 hours, and 24 minutes ago, and its protocols were started approximately 3 minutes before it booted. The switch was last configured by user **admin1** on May 27, 2009, and there is currently one user logged in to the switch.

The output also shows that the load average is 0.05 seconds for the last minute, 0.06 seconds for the last 5 minutes, and 0.01 seconds for the last 15 minutes.

Displaying the NTP Peers

Purpose

Verify that the time has been obtained from an NTP server.

Action

Enter the `show ntp associations` operational mode command to display the NTP server from switch obtained its time.

```
user@switch> show ntp associations
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*ntp.net .GPS.            1 u  414 1024  377    3.435    4.002   0.765
```

Meaning

The asterisk (*) in front of the NTP server name, or peer, indicates that the time is synchronized and obtained from this server. The delay, offset, and jitter are displayed in milliseconds.

Displaying the NTP Status

Purpose

View the configuration of the NTP server and the status of the system.

Action

Enter the `show ntp status` operational mode command to view the status of the NTP.

```
user@switch> show ntp status
status=0644 leap_none, sync_ntp, 4 events, event_peer/strat_chg,
version="ntpd 4.2.0-a Mon Apr 13 19:09:05 UTC 2009 (1)",
processor="powerpc", system="JUNOS9.5R1.8", leap=00, stratum=2,
precision=-18, rootdelay=2.805, rootdispersion=42.018, peer=48172,
```

```
refid=192.168.28.5,
reftime=cddd397a.60e6d7bf  Fri, Jun 12 2009 13:30:50.378, poll=10,
clock=cddd3b1b.ec5a2bb4  Fri, Jun 12 2009 13:37:47.923, state=4,
offset=3.706, frequency=-23.018, jitter=1.818, stability=0.303
```

Meaning

The output shows status information about the switch and the NTP.

NTP Authentication Keys

Time synchronization can be authenticated to ensure that the switch obtains its time services only from known sources. By default, network time synchronization is unauthenticated. The switch will synchronize to whatever system appears to have the most accurate time. We strongly encourage you to configure authentication of network time services.

To authenticate other time servers, include the `trusted-key` statement at the `[edit system ntp]` hierarchy level. The trusted keys refer to the configured key that is trusted and used by NTP for secure clock synchronization. Any configured key not referenced in the `trusted-key` is not qualified and is rejected by NTP. Only time servers that transmit network time packets containing one of the specified key numbers are eligible to be synchronized. Additionally, the key needs to match the value configured for that key number. Other systems can synchronize to the local switch without being authenticated.

```
[edit system ntp]
 trusted-key[ key-numbers ];
```

Each key can be any 32-bit unsigned integer except 0. Include the **key** option in the **peer**, **server**, or `broadcast` statements to transmit the specified authentication key when transmitting packets. The key is necessary if the remote system has authentication enabled so that it can synchronize to the local system.

To define the authentication keys, include the `authentication-key` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
authentication-key key-number type type value password;
```

number is the key number, *type* is the authentication type (only Message Digest 5 [MD5] , SHA1, and SHA2-256 are supported), and *password* is the password for this key. The key number, type, and password must match on all systems using that particular key for authentication. There must be no space in the password for configuring the Network Time Protocol (NTP) authentication-key.

Configure Devices to Listen to Broadcast Messages Using NTP

When you are using NTP, you can configure the local router or switch to listen for broadcast messages on the local network to discover other servers on the same subnet by including the `broadcast-client` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
broadcast-client;
```

When the router or switch detects a broadcast message for the first time, it measures the nominal network delay using a brief client-server exchange with the remote server. It then enters *broadcast client* mode, in which it listens for, and synchronizes to, succeeding broadcast messages.

To avoid accidental or malicious disruption in this mode, both the local and remote systems must use authentication and the same trusted key and key identifier.

Configure Devices to Listen to Multicast Messages Using NTP

When you are using NTP, you can configure the local router or switch to listen for multicast messages on the local network to discover other servers on the same subnet by including the `multicast-client` statement at the `[edit system ntp]` hierarchy level:

```
[edit system ntp]
multicast-client <address>;
```

When the router or switch receives a multicast message for the first time, it measures the nominal network delay using a brief client-server exchange with the remote server. It then enters *multicast client* mode, in which it listens for, and synchronizes to, succeeding multicast messages.

You can specify one or more IP addresses. (You must specify an address, not a hostname.) If you do, the router or switch joins those multicast groups. If you do not specify any addresses, the software uses `224.0.1.1`.

To avoid accidental or malicious disruption in this mode, both the local and remote systems must use authentication and the same trusted key and key identifier.

5
CHAPTER

Configuring SSH and Console Connection

Understanding FIPS Authentication Methods | 63

Configuring a System Login Message and Announcement | 64

Limiting the Number of User Login Attempts for SSH Sessions | 65

Configuring SSH on the Evaluated Configuration | 66

Understanding FIPS Authentication Methods

IN THIS SECTION

- Username and Password Authentication over the Console and SSH | 63
- Username and Public Key Authentication over SSH | 64

The Juniper Networks Junos operating system (Junos OS) running in FIPS mode of operation allows a wide range of capabilities for users, and authentication is role-based. The following types of role-based authentication are supported in the FIPS mode of operation:

- "Username and Password Authentication over the Console and SSH" on page 63
- "Username and Public Key Authentication over SSH" on page 64

Username and Password Authentication over the Console and SSH

In this authentication method, the user is requested to enter the username and password. The device enforces the user to enter a minimum of 10 characters password that is chosen from the 96 human-readable ASCII characters.

NOTE: The maximum password length is 20 characters.

In this method, the device enforces a timed access mechanism—for example, first two failed attempts to enter the correct password (assuming 0 time to process), no timed access is enforced. When the user enters the password for the third time, the module enforces a 5 second delay. Each failed attempt thereafter results in an additional 5 second delay above the previous failed attempt. For example, if the fourth failed attempt is a 10 second delay, then the fifth failed attempt is a 15 second delay, the sixth failed attempt is a 20 second delay, and the seventh failed attempt is a 25 second delay.

Therefore, this leads to a maximum of seven possible attempts in a 1 minute period for each getty active terminal. So, the best approach for the attacker would be to disconnect after 4 failed attempts, and wait for a new getty to be spawned. This would allow the attacker to perform roughly 9.6 attempts per minute (576 attempts per hour or 60 minutes). This would be rounded off to 9 attempts per minute, because there is no such thing as 0.6 attempts. Thus the probability of a successful random attempt is

1/9610, which is less than 1/1 million. The probability of a success with multiple consecutive attempts in a 1 minute period is 9/(9610), which is less than 1/100,000.

Username and Public Key Authentication over SSH

In SSH public key authentication, you provide the username and validate the ownership of the private key corresponding to the public key stored on the server. The device supports ECDSA (P-256, P-384, and P-521) and RSA (2048, 3072, and 4092 modulus bit length) key-types. The probability of a success with multiple consecutive attempts in a 1-minute period is 5.6e7/(2128).

NOTE: The ssh-rsa authentication method is one of the allowed algorithms in FIPS mode.

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration

Configuring a System Login Message and Announcement

A system login message appears before the user logs in and a system login announcement appears after the user logs in. By default, no login message or announcement is displayed on the device.

To configure a system login message, use the following command:

```
[edit]
user@host# set system login message login-message-banner-text
```

To configure system announcement, use the following command:

```
[edit]
user@host# set system login announcement  system-announcement-text
```

NOTE:

- If the message text contains any spaces, enclose it in quotation marks.
- You can format the message using the following special characters:
  - \n—New line
  - \t—Horizontal tab
  - \'—Single quotation mark
  - \"—Double quotation mark
  - \\—Backslash

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration

Limiting the Number of User Login Attempts for SSH Sessions

A remote administrator may login to a device through SSH. Administrator credentials are stored locally on the device. If the remote administrator presents a valid username and password, access to the TOE is granted. If the credentials are invalid, the TOE allows the authentication to be retried after an interval that starts after 1 second and increases exponentially. If the number of authentication attempts exceed the configured maximum, no authentication attempts are accepted for a configured time interval. When the interval expires, authentication attempts are again accepted.

You can configure the device to limit the number of attempts to enter a password while logging through SSH. Using the following command, the connection can be terminated if a user fails to login after a specified number of attempts:

```
[edit system login]
user@host# set retry-options  tries-before-disconnect <number>
```

Here, `tries-before-disconnect` is the number of times a user can attempt to enter a password when logging in. The connection closes if a user fails to log in after the number specified. The range is from 1 through 10, and the default value is 10.

You can also configure a delay, in seconds, before a user can try to enter a password after a failed attempt.

```
[edit system login]
user@host# set retry-options  backoff-threshold  <number>
```

Here, `backoff-threshold` is the threshold for the number of failed login attempts before the user experiences a delay in being able to enter a password again. Use the `backoff-factor` option to specify the length of the delay in seconds. The range is from 1 through 3, and the default value is 2 seconds.

In addition, the device can be configured to specify the threshold for the number of failed attempts before the user experiences a delay in entering the password again.

```
[edit system login]
user@host# set retry-options  backoff-factor  <number>
```

Here, `backoff-factor` is the length of time, in seconds, before a user can attempt to log in after a failed attempt. The delay increases by the value specified for each subsequent attempt after the threshold. The range is from 5 through 10, and the default value is 5 seconds.

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration

Configuring SSH on the Evaluated Configuration

SSH is an allowed remote management interface in the evaluated configuration. This topic describes how to configure SSH on the device.

1. Before you begin, log in with your root account on the device running Junos OS Release 22.2R1 and edit the configuration.

NOTE: The commands shown configure SSH to use all of the allowed cryptographic algorithms.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms.

```
[edit system services ssh]
user@host# set hostkey-algorithm ssh-ecdsa
```

NOTE: We recommend you to use the ecdsa-sha2-nistp256 hostkey algorithm to ensure Common Criteria compliance.

2. Specify the command to disable rsa-sha2-512 and rsa-sha2-256 hostkey algorithms.

```
[edit system services ssh]
user@host# set hostkey-algorithm no-ssh-rsa
```

NOTE: The set system services ssh hostkey-algorithm no-ssh-rsa command will disable the rsa-sha2-512, rsa-sha2-256, and ssh-rsa hostkey algorithms.

3. Specify the SSH key-exchange algorithms.

```
[edit system services ssh]
user@host#set key-exchange [ ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 Diffie-hellman-group14-sha1 ]
```

4. Specify all the permissible message authentication code algorithms.

```
[edit system services ssh]
user@host#set macs [ hmac-sha1 hmac-sha2-256 hmac-sha2-512 ]
```

5. Specify the ciphers allowed for protocol version 2.

```
[edit system services ssh]
user@host#set ciphers [ aes128-cbc aes128-ctr aes192-cbc aes192-ctr aes256-cbc aes256-ctr ]
```

6. (Optional step) Specify the number of minutes or maximum amount of data, before a rekey is forced on a session. The time limit must not be set greater than one hour and the data limit must not be set greater than one gigabyte.

```
[edit system services ssh]
user@host#set rekey time-limit minutes
user@host#set rekey data-limit bytes
```

RELATED DOCUMENTATION

- Understanding FIPS Authentication Methods | 63
- How to Enable and Configure Junos OS in FIPS Mode of Operation
- Limiting the Number of User Login Attempts for SSH Sessions | 65

CHAPTER 6

Configuring the Remote Syslog Server

Sample Syslog Server Configuration on a Linux System | 70

Forwarding Logs to the External Syslog Server | 77

Sample Syslog Server Configuration on a Linux System

IN THIS SECTION

- Configuring Event Logging to a Local File | 70
- Configuring Event Logging to a Remote Server | 71
- Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server | 71

A secure Junos OS environment requires auditing of events and storing them in a local audit file. The recorded events are simultaneously sent to an external syslog server. A syslog server receives the syslog messages streamed from the device. The syslog server must have an SSH client with NETCONF support configured to receive the streamed syslog messages.

The NDcPP logs capture the events, few of them are listed below:

- Committed changes
- Login and logout of users
- Failure to establish an SSH session
- Establishment or termination of an SSH session
- Changes to the system time

Configuring Event Logging to a Local File

You can configure storing of messages to a local file and the level of detail to be recorded with the `syslog` statement. This example stores logs in a file named messages:

```
[edit system]
syslog {
```

```
    file messages;
}
```

Configuring Event Logging to a Remote Server

Configure the export of audit information to a secure, remote server by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The following procedures show the configuration needed to send system log messages to a secure external server by using NETCONF over SSH.

Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server

The following procedure describes the steps to configure event logging to a remote server when the SSH connection to the TOE is initiated from the remote system log server.

1. Generate an RSA public key on the remote syslog server.

   ```
   $ ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
   ```

   You will be prompted to enter the desired passphrase. The storage location for the `syslog-monitor` key pair is displayed.

2. On the TOE, create a class named `monitor` that has permission to trace events.

   ```
   [edit]
   user@host# set system login class monitor permissions trace
   ```

3. Create a user named `syslog-mon` with the class monitor, and with authentication that uses the `syslog-monitor` key pair from the key pair file located on the remote syslog server.

   ```
   [edit]
   user@host# set system login user syslog-mon class monitor authentication ssh-rsa public key
   from syslog-monitor key pair
   ```

4. Set up NETCONF with SSH.

```
[edit]
user@host# set system services netconf ssh
```

5. Configure syslog to log all the messages at */var/log/messages*.

```
[edit]
user@host# set system syslog file messages any any
user@host# commit
```

6. On the remote system log server, start up the SSH agent. The start up is required to simplify the handling of the syslog-monitor key.

```
$ eval `ssh-agent`
```

7. On the remote syslog server, add the `syslog-monitor` key pair to the SSH agent.

```
$ ssh-add ~/.ssh/syslog-monitor
```

You will be prompted to enter the desired passphrase. Enter the same passphrase used in Step 1.

8. After logging in to the `external_syslog_server` session, establish a tunnel to the device and start NETCONF.

```
user@host# ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

9. After NETCONF is established, configure a system log events message stream. This RPC will cause the NETCONF service to start transmitting messages over the SSH connection that is established.

```
<rpc><get-syslog-events><stream>messages</stream></get-syslog-events></rpc>
```

10. The examples for syslog messages are listed below. Monitor the event log generated for admin actions on TOE as received on the syslog server. Examine the traffic that passes between the audit server and the TOE, observing that these data are not viewed during this transfer, and that they are successfully received by the audit server. Match the logs between local event and the remote event

logged in a syslog server and record the particular software (such as name, version, and so on) used on the audit server during testing.

The following output shows test log results for syslog server.

```
host@ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/host/.ssh/syslog-monitor.
Your public key has been saved in /home/host/.ssh/syslog-monitor.pub.
The key fingerprint is:
ef:75:d7:68:c5:ad:8d:6f:5e:7a:7e:9b:3d:f1:4d:3f syslog-monitor key pair
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|               ..|
|        S       +|
|         .     Bo|
|          . . *.X|
|         . . o E@|
|          .   .BX|
+-----------------+
[host@linux]$ cat /home/host/.ssh/syslog-monitor.pub
ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCrUREJUBpjwAoIgRrGy9zgt+
D2pikk3Q/Wdf8I5vr+njeqJhCx2bUAkrRbYXNILQQAZbg7kLfi/8TqqL
eon4HOP2e6oCSorKdx/GrOTzLONL4fh0EyuSAk8bs5JuwWNBUokV025
gzpGFsBusGnlj6wqqJ/sjFsMmfxyCkbY+pUWb8m1/A9YjOFT+6esw+9S
tF6Gbg+VpbYYk/Oday4z+z7tQHRFSrxj2G92aoliVDBLJparEMBc8w
LdSUDxmgBTM2oadOmm+kreBUQjrmr6775RJn9H9YwIxKOxGm4SFnX/Vl4
R+lZ9RqmKH2wodIEM34K0wXEHzAzNZ01oLmaAVqT
syslog-monitor key pair
[host@linux]$ eval `ssh-agent `
Agent pid 1453
[host@linux]$ ssh-add ~/.ssh/syslog-monitor
Enter passphrase for /home/host/.ssh/syslog-monitor:
Identity added: /home/host/.ssh/syslog-monitor (/home/host/.ssh/syslog-monitor)
```

Net configuration channel

```
host@linux]$ ssh syslog-mon@starfire -s netconf>test.out
host@linux]$ cat test.out
this is NDcPP test device

<!-- No zombies were killed during the creation of this user interface --
<!-- user syslog-mon, class j-monitor -><hello>
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:candidate:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:validate:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:url:1.0?protocol=http,ftp,file</
capability>
    <capability>http://xml.juniper.net/netconf/junos/1.0</capability>
    <capability>http://xml.juniper.net/dmi/system/1.0</capability>
  </capabilities>
  <session-id4129/session-id>
</hello>
]]>]]>
```

The following output shows event logs generated on the TOE that are received on the syslog server.

```
Jan 20 17:04:51  starfire sshd[4182]: error: Could not load host key: /etc/ssh/ssh_host_dsa_key
Jan 20 17:04:51  starfire sshd[4182]: error: Could not load host key: /etc/ssh/ssh_host_ecdsa_key
Jan 20 17:04:53  starfire sshd[4182]: Accepted password for sec-admin from 10.209.11.24 port
55571 ssh2
Jan 20 17:04:53  starfire mgd[4186]: UI_AUTH_EVENT: Authenticated user 'sec-admin' at permission
level 'j-administrator'
Jan 20 17:04:53  starfire mgd[4186]: UI_LOGIN_EVENT: User 'sec-admin' login, class 'j-
administrator' [4186], ssh-connection '10.209.11.24 55571 10.209.14.92 22', client-mode 'cli'
```

Net configuration channel

```
host@linux]$ ssh syslog-mon@starfire -s netconf
 this is NDcPP test device

<!-- No zombies were killed during the creation of this user interface --
<!-- user syslog-mon, class j-monitor -><hello>
```

```
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:candidate:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:validate:1.0</capability>
    <capability>urn:ietf:params:xml:ns:netconf:capability:url:1.0?protocol=http,ftp,file</
capability>
    <capability>http://xml.juniper.net/netconf/junos/1.0</capability>
    <capability>http://xml.juniper.net/dmi/system/1.0</capability>
  </capabilities>
  <session-id4129/session-id>
</hello>
]]>]]>
```

The following output shows that the local syslogs and remote syslogs received are similar.

```
Local : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd', PID 4317,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Dynamic
flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/dfcd', PID 4318,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/cfmd', PID 4319,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2ald', PID 4320,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2cpd'
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state machinesJan 20
```

```
17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2cpd', PID 4321,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Multicast
Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/mcsnoopd', PID
4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: commit
wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: start ffp
activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to
profiles.................................
```

```
Remote : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd', PID 4317,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Dynamic
flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/dfcd', PID 4318,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/cfmd', PID 4319,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2ald', PID 4320,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2cpd'
```

```
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state machinesJan 20
17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2cpd', PID 4321,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Multicast
Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/mcsnoopd', PID
4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: commit
wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: start ffp
activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to profiles ..............
```

Forwarding Logs to the External Syslog Server

When the device running Junos OS is set up for an external syslog server, the TOE forwards copies of local logs to the external syslog server and retains local copies of all logs when the TOE is configured in event log mode. In stream log mode, all logs except traffic logs are stored locally and can be forwarded to an external syslog server, whereas traffic logs can only be forwarded to an external syslog server.

The connection between the device running Junos OS and the syslog server is established on an event basis depending on preconfiguration of what type of logs are forwarded from local to external. When the configured condition is met, the device sends local logs to the external syslog server.

RELATED DOCUMENTATION

Sample Syslog Server Configuration on a Linux System | 70

CHAPTER 7

Configuring Audit Log Options

Configuring Audit Log Options in the Evaluated Configuration | 79

Sample Code Audits of Configuration Changes | 80

Configuring Audit Log Options in the Evaluated Configuration

IN THIS SECTION

- Configuring Audit Log Options for SRX5400, SRX5600, and SRX5800 Devices | 79

The following section describes how to configure audit log options in the evaluated configuration.

Configuring Audit Log Options for SRX5400, SRX5600, and SRX5800 Devices

To configure audit log options for SRX5400, SRX5600, and SRX5800 devices:

1. Specify the number of files to be archived in the system logging facility.

```
[edit system syslog]
root@host#set archive files 2
```

2. Specify the file in which to log data.

```
[edit system syslog]
root@host#set file syslog any any
```

3. Specify the size of files to be archived.

```
[edit system syslog]
root@host#set file syslog archive size 10000000
```

4. Log system messages in a structured format.

```
[edit system syslog]
root@host#set file syslog structured-data
```

5. Configure security log events in the audit log buffer.

```
[edit]
root@host#set security log cache
```

6. Specify how to process and export security logs.

```
[edit]
root@host#set security log mode event
```

RELATED DOCUMENTATION

Sample Code Audits of Configuration Changes | 80

Sample Code Audits of Configuration Changes

This sample code audits all changes to the configuration secret data and sends the logs to a file named **syslog:**

```
[edit system]
syslog {
    file syslog {
        authorization info;
        change-log info;
        interactive-commands info;
```

```
    }
}
```

This sample code expands the scope of the minimum audit to audit all changes to the configuration, not just secret data, and sends the logs to a file named **syslog:**

```
[edit system]
syslog {
    file syslog {
        any any;
        authorization info;
        change-log any;
        interactive-commands info;
        kernel info;
        pfe info;
    }
}
```

Example: System Logging of Configuration Changes

This example shows a sample configuration and makes changes to users and secret data. It then shows the information sent to the audit server when the secret data is added to the original configuration and committed with the `load` command.

```
[edit system]
location {
    country-code US;
    building B1;
}
...
login {
    message "UNAUTHORIZED USE OF THIS ROUTER\n\tIS STRICTLY PROHIBITED!";
        user admin {
            uid 2000;
             class super-user;
        authentication {
            encrypted-password “$ABC123”;
               # SECRET-DATA
        }
    }
    password {
        format md5;
```

```
    }
}
radius-server 192.0.2.15 {
    secret “$ABC123” # SECRET-DATA
}
services {
    ssh;
}
syslog {
    user *{
        any emergency;
    }
    file syslog {
        any notice;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
...
...
```

The new configuration changes the secret data configuration statements and adds a new user.

```
user@host# show | compare
[edit system login user admin authentication]
-    encrypted-password “$ABC123”; # SECRET-DATA
+    encrypted-password “$ABC123”; # SECRET-DATA
[edit system login]
+    user admin2 {
+        uid 2001;
+        class operator;
+        authentication {
+            encrypted-password “$ABC123”;
                    # SECRET-DATA
+        }
+     }
[edit system radius-server 192.0.2.15]
-    secret “$ABC123”; # SECRET-DATA
+    secret “$ABC123”; # SECRET-DATA
```

RELATED DOCUMENTATION

Configuring Audit Log Options in the Evaluated Configuration | 79

8
CHAPTER

Configuring Event Logging

Event Logging Overview | 85

Interpreting Event Messages | 90

Logging Changes to Secret Data | 91

Login and Logout Events Using SSH | 93

Logging of Audit Startup | 94

Event Logging Overview

The evaluated configuration requires the auditing of configuration changes through the system log.

In addition, Junos OS can:

- Send automated responses to audit events (syslog entry creation).
- Allow authorized managers to examine audit logs.
- Send audit files to external servers.
- Allow authorized managers to return the system to a known state.

The logging for the evaluated configuration must capture the events. The logging events are listed below:

Table 4 on page 85 shows sample for syslog auditing for NDcPPv2:

Table 4: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FAU_STG.1	None	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None

Table 4: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_RBG_EXT.1	None	None
FDP_RIP.2	None	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
FMT_MTD.1/CoreData	All management activities of TSF data	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None

Table 4: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FPT_STM.1	Discontinuous changes to time - either Administrator actuated or changed through an automated process.	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (such as, IP address).
FTA_SSL_EXT.1	The termination of a local interactive session by the session locking mechanism.	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.

Table 4: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None
FMT_MOF.1/Services	Starting and stopping of services.	None
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses. Source and destination ports. Transport Layer Protocol TOE Interface.
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets. Identifier of rule causing packet drop.

Table 4: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents
FFW_RUL_EXT.2	None	None
FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment.
FIA_X509_EXT.1	Session establishment with CA	Entire packet contents of packets transmitted/received during session establishment.
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses. Source and destination ports. Transport Layer Protocol TOE Interface.
	Indication of packets dropped due to too much network traffic.	TOE interface that is unable to process packets.

In addition, Juniper Networks recommends that logging also:

- Capture all changes to the configuration.
- Store logging information remotely.

RELATED DOCUMENTATION

Interpreting Event Messages | 90

Interpreting Event Messages

The following output shows a sample event message.

```
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-
server 1.2.3.4 secret]
```

Table 5 on page 90 describes the fields for an event message. If the system logging utility cannot determine the value in a particular field, a hyphen (-) appears instead.

Table 5: Fields in Event Messages

Field	Description	Examples
timestamp	Time when the message was generated, in one of two representations: • *MMM-DD HH:MM:SS.MS+/-HH:MM,* is the month, day, hour, minute, second and millisecond in local time. The hour and minute that follows the plus sign (+) or minus sign (-) is the offset of the local time zone from Coordinated Universal Time (UTC). • *YYYY-MM-DDTHH:MM:SS.MSZ* is the year, month, day, hour, minute, second and millisecond in UTC.	`Jul 24 17:43:28` is the timestamp expressed as local time in the United States. `2012-07-24T09:17:15.719Z` is `9:17 AM UTC on 24 July 2012.`
hostname	Name of the host that originally generated the message.	`router1`
process	Name of the Junos OS process that generated the message.	`mgd`
processID	UNIX process ID (PID) of the Junos OS process that generated the message.	`4153`
TAG	Junos OS system log message tag, which uniquely identifies the message.	`UI_DBASE_LOGOUT_EVENT`

Table 5: Fields in Event Messages *(Continued)*

Field	Description	Examples
username	Username of the user initiating the event.	"admin"
message-text	English-language description of the event .	set: [system radius-server 1.2.3.4 secret]

RELATED DOCUMENTATION

Event Logging Overview | 85

Logging Changes to Secret Data

The following are examples of audit logs of events that change the secret data.

Load Merge

When a `load merge` command is issued to merge the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-
server 1.2.3.4 secret]
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user admin authentication encrypted-password]
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user admin2 authentication encrypted-password]
```

Load Replace

When a `load replace` command is issued to replace the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system
radius-server 1.2.3.4 secret]
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system login
user admin authentication encrypted-password]
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system login
user admin authentication encrypted-password]
```

Load Override

When a `load override` command is issued to override the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 25 14:25:51  router1 mgd[4153]: UI_LOAD_EVENT: User 'admin' is performing a 'load override'
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' override: CC_config2.txt
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-
server 1.2.3.4 secret]
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user admin authentication encrypted-password]
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user admin authentication encrypted-password]
```

Load Update

When a `load update` command is issued to update the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 25 14:31:03  router1 mgd[4153]: UI_LOAD_EVENT: User 'admin' is performing a 'load update'
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' update: CC_config2.txt
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-
server 1.2.3.4 secret]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system radius-
server 1.2.3.4 secret] ""
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user admin authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system login
user admin authentication encrypted-password] ""
```

```
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login
user test authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system login
user test authentication encrypted-password] ""
```

For more information about configuring parameters and managing log files, see the *Junos OS System Log Messages Reference*.

RELATED DOCUMENTATION

Forwarding Logs to the External Syslog Server | 77

Interpreting Event Messages | 90

Login and Logout Events Using SSH

System log messages are generated whenever a user successfully or unsuccessfully attempts SSH access. Logout events are also recorded. For example, the following logs are the result of two failed authentication attempts, then a successful one, and finally a logout:

```
Dec 20 23:17:35  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:42  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:53  bilbo sshd[16645]: Accepted password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:53  bilbo mgd[16648]: UI_AUTH_EVENT: Authenticated user 'op' at permission level
                                   'j-operator'
Dec 20 23:17:53  bilbo mgd[16648]: UI_LOGIN_EVENT: User 'op' login, class 'j-operator' [16648]
Dec 20 23:17:56  bilbo mgd[16648]: UI_CMDLINE_READ_LINE: User 'op', command 'quit '
Dec 20 23:17:56  bilbo mgd[16648]: UI_LOGOUT_EVENT: User 'op' logout
```

RELATED DOCUMENTATION

Interpreting Event Messages | 90

Logging of Audit Startup

The audit information logged includes startups of Junos OS. This in turn identifies the startup events of the audit system, which cannot be independently disabled or enabled. For example, if Junos OS is restarted, the audit log contains the following information:

```
Dec 20 23:17:35  bilbo syslogd: exiting on signal 14
Dec 20 23:17:35  bilbo syslogd: restart
Dec 20 23:17:35  bilbo syslogd /kernel: Dec 20 23:17:35 init: syslogd (PID 19128) exited with
status=1
Dec 20 23:17:42  bilbo /kernel:
Dec 20 23:17:53  init: syslogd (PID 19200) started
```

RELATED DOCUMENTATION

Login and Logout Events Using SSH | 93

9

CHAPTER

Configuring a Secure Logging Channel

Creating a Secure Logging Channel | 96

Creating a Secure Logging Channel

IN THIS SECTION

- Configuring a Trusted Path or Channel Between a Device Running Junos OS and a Remote External Storage Server | 97

This section describes how to place the device in an evaluated configuration to provide an encrypted communication channel over an IPsec VPN tunnel, between a device running Junos OS and a remote external storage server (syslog server).

NOTE: The ssh-rsa authentication method is one of the allowed algorithms in FIPS mode.

Table 6 on page 96 lists all the supported algorithms for the IPsec VPN tunnel.

Table 6: IPsec VPN Tunnel Supported Algorithms

IKE Phase1 Proposal			
Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
pre-shared-keys rsa-signatures-2048 ecdsa-signatures-256 ecdsa-signatures-384	sha-256 sha-384	group14 group19 group20 group24	aes-128-cbc aes-128-gcm aes-192-cbc aes-256-cbc aes-256-gcm

IPSec Phase2 Proposal			
Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
hmac-sha1-96 hmac-sha-256-128	group14 group19 group20 group24	ESP	aes-128-cbc aes-128-gcm aes-192-cbc aes-192-gcm aes-256-cbc aes-256-gcm

Configuring a Trusted Path or Channel Between a Device Running Junos OS and a Remote External Storage Server

This section describes the configuration details required to provide an encrypted communication channel between a device running Junos OS and the remote external storage server through an IPsec VPN tunnel.

NOTE: The remote external storage server is a Linux-based syslog server on which the IPsec VPN Tunnel is terminated at the outbound interface Eth1. The log data transferred from the device is sent to the syslog termination interface Eth2 and the StrongSwan application to provide the IPsec VPN capability.

Table 7 on page 98 lists the IPsec VPN tunnel details used in this example.

Table 7: IPsec VPN Tunnel Information

Phase 1 Proposal (P1, IKE)				Phase 2 Proposal (P2, IPSec)			
Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm	Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
pre-shared-keys	sha-256	group14	aes-128-cbc	hmac-sha1-96	group14	ESP	aes-128-cbc

Figure 1 on page 98 illustrates the encrypted communication channel between a device running Junos OS and a remote external storage server. An IPsec tunnel is established between a devices egress interface (Intf-1) and a remote syslog server outbound interface (Eth1). Data is then forwarded internally on the remote external storage server from its outbound interface Eth1; that is, the VPN endpoint to Eth2.

Figure 1: IPsec VPN Tunnel

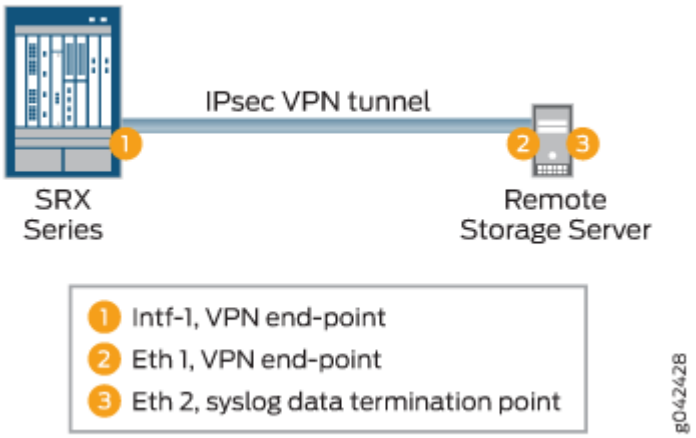

Table 8 on page 99 provides the interface and IP configuration details used in this example.

Table 8: Interface and IP Configuration Details for the Trusted Path

Device Running Junos OS	Remote Storage Server
IP Address: "Intf-2" interface: GE-0/0/1 – IP Address: 198.51.100.2 "Intf-1" interface: GE-0/0/2 - IP Address: 198.51.100.1 Enable: Syslog logging to remote syslog server	IP Address: Eth1: 198.51.100.3 Eth2: 203.0.113.1 Gateway Eth1: 198.51.100.1 Tools: SSH and Strongswan (for IPsec VPN)

To configure the trusted path or channel between a device running Junos OS and a remote external storage server:

1. Enable stream logging for traffic logs.

```
[edit security]
user@host#set log cache
user@host#set log mode event
user@host#set log source-address 198.51.100.2
user@host#set log stream STREAM category all
user@host#set log stream STREAM host 203.0.113.1
```

NOTE: `192.168.2.1` is the IP address of the syslog server outbound interface at which the IPsec VPN tunnel is terminated, and `20.20.20.2` is the IP address of the syslog server interface for which log data is destined.

2. Enable syslog on the device.

```
[edit system]
user@host#set syslog user * any emergency
user@host#set syslog host 203.0.113.1 any any
user@host#set syslog file SYSLOG any any
user@host#set syslog file SYSLOG authorization info
user@host#set syslog file SYSLOG_COMMANDS interactive-commands error
user@host#set syslog file traffic-log any any
```

```
user@host#set syslog file traffic-log match RT_FLOW_SESSION
user@host#set syslog source-address 198.51.100.2
```

3. Enable VPN on the device.

   IKE setup:

```
[edit security]
user@host#set ike proposal IKE_Proposal authentication-method pre-shared-keys
user@host#set ike proposal IKE_Proposal dh-group group14
user@host#set ike proposal IKE_Proposal authentication-algorithm sha-256
user@host#set ike proposal IKE_Proposal encryption-algorithm aes-128-cbc
user@host#set ike policy IKE_Policy mode main
user@host#set ike policy IKE_Policy proposals IKE_Proposal
user@host#set ike policy IKE_Policy pre-shared-key ascii-text 12345
user@host#set ike gateway GW ike-policy IKE_Policy
user@host#set ike gateway GW address 198.51.100.3
user@host#set ike gateway GW local-identity inet 198.51.100.1
user@host#set ike gateway GW external-interface ge-0/0/2
user@host#set ike gateway GW version v2-only
```

   IPsec setup:

```
[edit security ipsec]
user@host#set proposal IPsec_Proposal protocol esp
root@host#set proposal IPsec_Proposal authentication-algorithm hmac-sha1-96
root@host#set proposal IPsec_Proposal encryption-algorithm aes-128-cbc
root@host#set policy IPsec_Policy perfect-forward-secrecy keys group14
root@host#set policy IPsec_Policy proposals IPsec_Proposal
root@host#set vpn VPN bind-interface st0.0
root@host#set vpn VPN ike gateway GW
root@host#set vpn VPN ike ipsec-policy IPsec_Policy
root@host#set vpn VPN establish-tunnels immediately
```

4. Perform the following additional configurations on the device.

   IKE trace log:

```
[edit security ike]
root@host#set traceoptions file IKE_Trace
```

```
root@host#set traceoptions file size 10000000
root@host#set ike traceoptions flag all
```

Flow trace:

```
[edit security flow ]
root@host#set traceoptions file DEBUG
root@host#set traceoptions file size 1000000
root@host#set traceoptions flag all
```

Route options:

```
[edit ]
root@host#set routing-options static route 203.0.113.2/24 qualified-next-hop st0.0 preference
1
```

Address book configuration:

```
[edit security address-book]
root@host#set global address trustLAN 198.51.100.0/24
root@host#set global address unTrustLAN 198.51.100.3/24
```

Zone configuration:

```
[edit security zones]
root@host#set security-zone trustZone host-inbound-traffic system-services all
root@host#set security-zone trustZone host-inbound-traffic protocols all
root@host#set security-zone trustZone interfaces ge-0/0/1.0
root@host#set security-zone unTrustZone host-inbound-traffic system-services all
root@host#set security-zone unTrustZone host-inbound-traffic protocols all
root@host#set security-zone unTrustZone interfaces st0.0
root@host#set security-zone unTrustZone interfaces ge-0/0/2.0
```

Policy configuration:

```
[edit security policies]
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match source-address
trustLAN
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match destination-
```

```
address unTrustLAN
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match application any
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then permit
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then log session-init
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then log session-close
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match source-address
unTrustLAN
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match destination-
address trustLAN
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match application any
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then permit
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then log session-init
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then log session-close
```

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration

Sample Syslog Server Configuration on a Linux System | 70

10
CHAPTER

Configuring VPNs

Configuring VPN on a Device Running Junos OS | 104

Configuring VPN on a Device Running Junos OS

IN THIS SECTION

- Configuring an IPsec VPN with a Preshared Key for IKE Authentication | 106
- Configuring an IPsec VPN with an RSA Signature for IKE Authentication | 114
- Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication | 118

This section describes sample configurations of an IPsec VPN on a Junos OS device using the following IKE authentication methods:

- "Configuring an IPsec VPN with a Preshared Key for IKE Authentication" on page 106
- "Configuring an IPsec VPN with an RSA Signature for IKE Authentication" on page 114
- "Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication" on page 118

Figure 2 on page 104 illustrates the VPN topology used in all the examples described in this section. Here, H0 and H1 are the host PCs, R0 and R2 are the two endpoints of the IPsec VPN tunnel, and R1 is a router to route traffic between the two different networks.

NOTE: The router R1 can be a Linux-based router, a Juniper Networks device, or any other vendor router.

Figure 2: VPN Topology

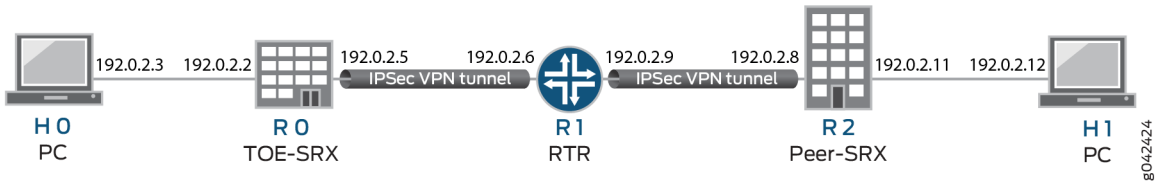

Table 9 on page 105 provides a complete list of the supported IKE protocols, tunnel modes, Phase 1 negotiation mode, authentication method or algorithm, encryption algorithm, DH groups supported for the IKE authentication and encryption (Phase1, IKE Proposal), and for IPsec authentication and

encryption (Phase2, IPsec Proposal). The listed protocols, modes, and algorithms are supported and required for 22.2R1 Common Criteria.

Table 9: VPN Combination Matrix

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	pre-shared-keys	sha-256	group14	
IKEv2			rsa-signatures-2048	sha-384	group19	aes-128-cbc
			ecdsa-signatures-256		group20	aes-128-gcm
			ecdsa-signatures-384		group24	aes-192-cbc
						aes-256-cbc
						aes-256-gcm

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha1-96	group14	ESP	
IKEv2			hmac-sha-256-128	group19		aes-128-cbc
				group20		aes-128-gcm
				group24		aes-192-cbc

(Continued)

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
						aes-192-gcm
						aes-256-cbc
						aes-256-gcm

NOTE: The following sections provide sample configurations of IKEv1 IPsec VPN examples for selected algorithms. Authentication algorithms can be replaced in the configurations to accomplish the user's desired configurations. Use `set security ike gateway <gw-name> version v2-only` command for IKEv2 IPsec VPN.

Configuring an IPsec VPN with a Preshared Key for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using a preshared key as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 10 on page 106

Table 10: IKE or IPsec Authentication

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	pre-shared-keys	sha-256	group14	aes-256-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha-256-128	group14	ESP	aes-256-cbc

NOTE: A device running Junos OS uses certificate-based authentication or preshared keys for IPsec. TOE accepts ASCII preshared or bit-based keys up to 255 characters (and their binary equivalents) that contain uppercase and lowercase letters, numbers, and special characters such as !, @, #, $, %, ^, &, *, (, and). The device accepts the preshared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges. The Junos OS does not impose minimum complexity requirements for preshared keys. Hence, users are advised to carefully choose long preshared keys of sufficient complexity.

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Initiator

To configure the IPsec VPN with preshared key IKE authentication on the initiator:

1. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method pre-shared-keys
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha256
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, `ike-proposal1` is the IKE proposal name given by the authorized administrator.

2. Configure the IKE policy.

```
[edit]
user@host# set security ike policy ike-policy1 mode main
user@host# set security ike policy ike-policy1 proposals ike-proposal1
```

NOTE: Here, `ike-policy1` is the IKE policy name and `ike-proposal1` is the IKE proposal name given by the authorized administrator.

```
user@host# prompt security ike policy ike-policy1 pre-shared-key ascii-text
New ascii-text (secret):
Retype new ascii-text (secret):
```

NOTE: You must enter and reenter the preshared key when prompted. For example, the preshared key can be *CertSqa@jnpr2014*.

NOTE: The preshared key can alternatively be entered in hexadecimal format. For example:

```
[edit]
root@host# prompt security ike policy ike-policy1 pre-shared-key hexadecimal
New hexadecimal (secret):
Retype new hexadecimal (secret):
```

Enter the hexadecimal preshared key value.

3. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set security proposal ipsec-proposal1 protocol esp
user@host# set security proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set security proposal ipsec-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

4. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set security policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set security policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, `ipsec-policy1` is the IPsec policy name and `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

5. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface ge-0/0/2
```

NOTE: Here, `gw1` is an IKE gateway name, `192.0.2.8` is the peer VPN endpoint IP, `192.0.2.5` is the local VPN endpoint IP, and `ge-0/0/2` is a local outbound interface as the VPN endpoint. The following additional configuration is also needed in the case of IKEv2

```
[edit security ike]
user@host# set gateway gw1 version v2-only
```

6. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0 preference
1
```

NOTE: Here, `vpn1` is the VPN tunnel name given by the authorized administrator.

7. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

8. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

9. Commit your configuration.

```
user@host# commit
```

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Responder

To configure the IPsec VPN with preshared key IKE authentication on the responder:

1. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method pre-shared-keys
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha256
user@host# set proposal ike-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, `ike-proposal1` is the IKE proposal name given by the authorized administrator.

2. Configure the IKE policy.

```
[edit]
user@host# set security ike policy ike-policy1 mode main
user@host# set security ike policy ike-policy1 proposals ike-proposal1
```

NOTE: Here, `ike-policy1` is the IKE policy name and `ike-proposal1` is the IKE proposal name given by the authorized administrator.

```
user@host# prompt security ike policy ike-policy1 pre-shared-key ascii-text
New ascii-text (secret):
Retype new ascii-text (secret):
```

NOTE: You must enter and reenter the preshared key when prompted. For example, the preshared key can be *CertSqa@jnpr2014*.

NOTE: The pre-share key could alternatively be entered in hexadecimal format. For example,

```
user@host# prompt security ike policy ike-policy1 pre-shared-key hexadecimal
New hexadecimal (secret):
Retype new hexadecimal (secret):
```

Here, the hexadecimal preshared key can be `cc2014bae9876543`.

3. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

4. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, `ipsec-policy1` is the IPsec policy name and `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

5. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.5
user@host# set gateway gw1 local-identity inet 192.0.2.8
user@host# set gateway gw1 external-interface ge-0/0/2
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.5 is the peer VPN endpoint IP, 192.0.2.8 is the local VPN endpoint IP, and ge-0/0/2 is a local outbound interface as the VPN endpoint. The following additional configuration is also needed in the case of IKEv2.

```
[edit security ike]
user@host# set gateway gw1 version v2-only
```

6. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.7/24 qualified-next-hop st0.0 preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

7. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

8. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

9. Commit your configuration.

```
user@host# commit
```

Configuring an IPsec VPN with an RSA Signature for IKE Authentication

The following section provides an example to configure Junos OS devices for IPsec VPN using RSA Signature as IKE Authentication method, whereas, the algorithms used in IKE/IPsec authentication/ encryption is as shown in the following table. In this section, you configure devices running Junos OS for IPsec VPN using an RSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 11 on page 114.

Table 11: IKE/IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	rsa-signatures-2048	sha-256	group19	aes-128-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha-256-128	group19	ESP	aes-128-cbc

Configuring IPsec VPN with RSA Signature as IKE Authentication on the Initiator or Responder

To configure the IPsec VPN with RSA signature IKE authentication on the initiator:

1. Configure the PKI. See Example: Configuring PKI.
2. Generate the RSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load the CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load the CRL. See Example: Manually Loading a CRL onto the Device .
5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.
6. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method rsa-signatures
user@host# set proposal ike-proposal1 dh-group group19
user@host# set proposal ike-proposal1 authentication-algorithm sha-256
user@host# set proposal ike-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, `ike-proposal1` is the name given by the authorized administrator.

7. Configure the IKE policy.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

NOTE: Here, `ike-policy1` IKE policy name given by the authorized administrator.

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, `ipsec-proposal1` is the name given by the authorized administrator.

9. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group19
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, `ipsec-policy1` is the name given by the authorized administrator.

10. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface fe-0/0/1
```

NOTE: Here, `192.0.2.8` is the peer VPN endpoint IP, `192.0.2.5` is the local VPN endpoint IP, and `fe-0/0/1` is the local outbound interface as VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gateway gw1 version v2-only
```

11. Configure VPN.

```
[edit security ipsec]
user@host# set vpn vpn1 ike gateway gw1
user@host# set vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set vpn vpn1 bind-interface st0.0
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

```
[edit]
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0
preference 1
```

12. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zone and trustLan and untrustLan are preconfigured network addresses.

13. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-
close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

14. Commit the configuration.

```
[edit]
user@host# commit
```

Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using an ECDSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption are shown in Table 12 on page 119.

Table 12: IKE or IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	ecdsa-signatures-256	sha-384	group14	aes-256-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	No Algorithm	group14	ESP	aes-256-gcm

Configuring IPsec VPN with ECDSA signature IKE authentication on the Initiator

To configure the IPsec VPN with ECDSA signature IKE authentication on the initiator:

1. Configure the PKI. See, Example: Configuring PKI.
2. Generate the ECDSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load CRL. See Example: Manually Loading a CRL onto the Device .
5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.
6. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method ecdsa-signatures-256
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha-384
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, `ike-proposal1` is the IKE proposal name given by the authorized administrator.

7. Configure the IKE policy.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-256-gcm
```

NOTE: Here, `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

9. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, `ipsec-policy1` is the IPsec policy name and `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

10. Configure IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
```

```
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface ge-0/0/2
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.8 is the peer VPN endpoint IP, 192.0.2.5 is the local VPN endpoint IP, and ge-0/0/2 is a local outbound interface as the VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gateway gw1 version v2-only
```

11. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0
preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

12. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-
close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

13. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-
close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

14. Commit your configuration.

```
user@host# commit
```

Configuring IPsec VPN with ECDSA signature IKE authentication on the Responder

To configure IPsec VPN with ECDSA signature IKE authentication on the responder:

1. Configure the PKI. See, Example: Configuring PKI.
2. Generate the ECDSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load the CRL. See Example: Manually Loading a CRL onto the Device .

5. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method ecdsa-signatures-256
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha-384
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, `ike-proposal1` is the IKE proposal name given by the authorized administrator.

6. Configure the IKE policy.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

7. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-256-gcm
```

NOTE: Here, `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

8. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, `ipsec-policy1` is the IPsec policy name and `ipsec-proposal1` is the IPsec proposal name given by the authorized administrator.

9. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.5
user@host# set gateway gw1 local-identity inet 192.0.2.8
user@host# set gateway gw1 external-interface ge-0/0/1
```

NOTE: Here, `gw1` is an IKE gateway name, `192.0.2.5` is the peer VPN endpoint IP, `192.0.2.8` is the local VPN endpoint IP, and `ge-0/0/1` is a local outbound interface as the VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gateway gw1 version v2-only
```

10. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.1/24 qualified-next-hop st0.0
preference 1
```

NOTE: Here, `vpn1` is the VPN tunnel name given by the authorized administrator.

11. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
```

```
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-
close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

12. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-
close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

13. Commit your configuration.

```
user@host# commit
```

RELATED DOCUMENTATION

Sample Syslog Server Configuration on a Linux System | 70

Understanding a Security Flow Policy on a Device Running Junos OS | 128

CHAPTER 11

Configuring Security Flow Policies

Understanding a Security Flow Policy on a Device Running Junos OS | 128

Understanding a Security Flow Policy on a Device Running Junos OS

IN THIS SECTION

- Configuring a Security Flow Policy in Firewall Bypass Mode | 128
- Configuring a Security Policy in Firewall Discard Mode | 129
- Configuring a Security Flow Policy in IPsec Protect Mode | 130

You can define a security flow policy on a device running Junos OS to inspect and process network packets. The device can permit, deny, and log operations to be associated with each policy. Each of these policies are associated to zones on which distinct network interfaces are bound.

The following modes can be defined for a security flow policy to determine how a device directs traffic:

- Bypass—The `Permit` option directs the traffic traversing the device through the stateful firewall inspection, but not through the IPsec VPN tunnel.
- Discard—The `Deny` option inspects and drops all packets that do not match any `Permit` policies.
- Protect—The traffic is routed through an IPsec tunnel based on the combination of route lookup and `Permit` policy inspection.
- Log—This option logs traffic and session information for all the modes mentioned above.

The following sections describe how to configure a security policy for each of these modes:

- "Configuring a Security Flow Policy in Firewall Bypass Mode" on page 128
- "Configuring a Security Policy in Firewall Discard Mode" on page 129
- "Configuring a Security Flow Policy in IPsec Protect Mode" on page 130

Configuring a Security Flow Policy in Firewall Bypass Mode

To configure a security flow policy for firewall bypass mode:

- Configure the security policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses. `junos-ssh` is an example of a Junos OS default predefined application that can be configured in a security policy to enforce SSH traffic.

Configuring a Security Policy in Firewall Discard Mode

To configure a security flow policy for firewall discard mode:

- Configure the security policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match source-address
untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match destination-
address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match application junos-
telnet
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then deny
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then session-close
```

NOTE: Here, `trustZone` and `untrustZone` are the preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses. `junos-telnet` is an example of a Junos OS

default predefined application that can be configured in a security policy to enforce Telnet traffic.

Configuring a Security Flow Policy in IPsec Protect Mode

To configure a security flow policy for IPSec protect mode:

1. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 198.51.100.14/24 qualified-next-hop st0.0
preference 1
```

NOTE: Here, `gw1` and `ipsec-policy1` are preconfigured IKE and IPsec policies.

2. Configure the security policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

RELATED DOCUMENTATION

Configuring VPN on a Device Running Junos OS

Configuring VPN on a Device Running Junos OS | **104**

CHAPTER 12

Configuring Traffic Filtering Rules

Overview | 133

Understanding Protocol Support | 133

Configuring Traffic Filter Rules | 135

Configuring Default Deny-All and Reject Rules | 136

Logging the Dropped Packets Using Default Deny-all Option | 137

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets | 138

Configuring Default Reject Rules for Source Address Spoofing | 139

Configuring Default Reject Rules with IP Options | 140

Configuring Default Reject Rules | 141

Overview

By default, the TOE denies all traffic through an SRX Series device. In fact, an implicit default security policy exists that denies all packets. You can change this behavior by configuring a standard security policy that permits certain types of traffic. The implicit default policy can be changed to permit all traffic with the `set security policies default-policy` command; however, this is not recommended.

The security policy rule set is an ordered list of security policy entries enforced by the firewall rules, each of which contains the specification of a network flow and an action:

- Source IP address and network mask
- Destination IP address and network mask
- Protocol
- Source port
- Destination port
- Action: permit, deny, drop silently, log

Each packet is compared against entries in the security policy rule set in sequential order until one is found that matches the specification in the policy, or until the end of the rule set is reached, in which case the implicit default policy is implemented and the packet is discarded.

RELATED DOCUMENTATION

Reordering Security Policies

Understanding Protocol Support

You can configure the devices running Junos OS to perform stateful network traffic filtering on network packets using network traffic protocols and network fields as described in Table 13 on page 134.

Table 13: Network Traffic Protocols and Fields

Protocol or RFC	Fields
ICMPv4 - RFC 792, Internet Control Message Protocol version 4	• Type • Code
ICMPv6 - RFC 4443, Internet Control Message Protocol version 6	• Type • Code
IPv4 - RFC 791, Internet Protocol	• Source address • Destination address • Transport Layer Protocol
IPv6 - RFC 2460, Internet Protocol	• Source address • Destination address • Transport Layer Protocol
TCP - RFC 793, Transmission Control Protocol	• Source port • Destination port
UDP - RFC 768, User Datagram Protocol	• Source port • Destination port

The following protocols are also supported on devices running Junos OS and are a part of this evaluation.

- IPsec
- IKE
- SSH

The following protocols are supported on devices running Junos OS but are not included in the scope of this evaluation.

- OSPF
- BGP
- RIP

RELATED DOCUMENTATION

Configuring Traffic Filter Rules | 135

Configuring Traffic Filter Rules

Traffic filter rules can be configured on a device to enforce validation against protocols attributes and direct traffic accordingly to the configured attributes. These rules are based on zones on which network interfaces are bound.

The following procedure describes how to configure traffic filter rules to direct FTP traffic from source `trustZone` to destination `untrustZone` and from source network `trustLan` to destination network `untrustLan`. Here, traffic is traversing from the devices interface A on `trustZone` to interface B on `untrustZone`.

1. Configure a zone and its interfaces.

```
[edit]
user@host# set security zones security-zone trustLan interfaces ge-0/0/0
```

2. Configure the security policy in the specified zone-to-zone direction and specify the match criteria.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match source-address
trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match destination-
address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match application ftp
```

3. Configure the security policy in the specified zone-to-zone direction and specify the action to take when a packet matches a criteria.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then session-close
```

NOTE: Here, `trustZone` and `untrustZone` are preconfigured security zones and `trustLan` and `untrustLan` are preconfigured network addresses.

RELATED DOCUMENTATION

Understanding Protocol Support | 133

Configuring Default Deny-All and Reject Rules

By default, security devices running Junos OS deny traffic unless rules are explicitly created to allow it using the following command:

```
[edit]
user@host#set security policies default-policy deny-all
```

You can configure your security devices running Junos OS to enforce the following default reject rules with logging on all network traffic:

- Invalid fragments
- Fragmented IP packets that cannot be reassembled completely
- Where the source address is equal to the address of the network interface
- Where the source address does not belong to the networks associated with the network interface
- Where the source address is defined as being on a broadcast network
- Where the source address is defined as being on a multicast network

- Where the source address is defined as being a loopback address
- Where the source address is a multicast packet
- Where the source or destination address is a link-local address
- Where the source or destination address is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4
- Where the source or destination address is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6
- With the IP option Loose Source Routing, Strict Source Routing, or Record Route is specified

Logging the Dropped Packets Using Default Deny-all Option

The evaluated configuration device drops all IPv6 traffic by default. This topic describes how to log packets dropped by this default deny-all option.

1. Before you begin, log in with your root account on a Junos OS device running Junos OS Release 22.2R1 and edit the configuration.

> **NOTE**: You can enter the configuration commands in any order and commit all the commands at once.

To log packets dropped by the default deny-all option:

1. Configure a network security policy in a global context and specify the security policy match criteria.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log match source-address any
destination-address any application any
```

2. Specify the policy action to take when the packet matches the criteria.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log then deny
```

3. Configure the security policy to enable logs at the session initialization time.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log then log session-init
```

NOTE: This procedure might capture a very large amount of data until you have configured the other policies.

To permit all IPv6 traffic into an SRX Series device, configure the device with flow-based forwarding mode. While the default policy in flow-based forwarding mode is still to drop all IPv6 traffic, you can now add rules to permit selected types of IPv6 traffic.

```
user@host# set security forwarding-options family inet6 mode flow-based
```

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets

This topic describes how to configure mandatory reject rules for invalid fragments and fragmented IP packets that cannot be reassembled.

1. Before you begin, log in with your root account on a Junos OS device running Junos OS Release 22.2R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure mandatory reject rules:

1. Specify the flow configuration to forcefully reassemble the IP fragments.

```
[edit]
user@host# set security flow force-ip-reassembly
```

2. Delete the screen ID and the IDS options and enable the ICMP fragment IDS option.

```
[edit]
user@host# delete security screen ids-option trustScreen icmp fragment
```

3. Delete the IP layer IDS option and enable the IP fragment blocking IDS option.

```
[edit]
user@host# delete security screen ids-option trustScreen ip block-frag
```

Configuring Default Reject Rules for Source Address Spoofing

The following guidelines describe when to configure the default reject rules for source address spoofing:

- When the source address is equal to the address of the network interface where the network packet was received.
- When the source address does not belong to the networks associated with the network interface where the network packet was received.
- When the source address is defined as being on a broadcast network.

1. Before you begin, log in with your root account on a Junos OS device running Junos OS Release 22.2R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure default reject rules to log source address spoofing:

1. Configure the security screen features and enable the IP address spoofing IDS option.

```
[edit]
user@host# set security screen ids-option trustScreen ip spoofing
```

2. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit]
user@host# set security zones security-zone trustZone screen trustScreen
```

Configuring Default Reject Rules with IP Options

This topic describes how to configure default reject rules with IP options. The IP options enable the device to either block any packets with loose or strict source route options or detect such packets and then record the event in the counters list for the ingress interface.

1. Before you begin, log in with your root account to an SRX Series device running Junos OS Release 22.2R1.

> **NOTE**: You can enter the configuration commands in any order and commit all the commands at once.

To configure the default reject rules with IP options:

1. Configure the screen features to enable IP options.

```
[edit security screen ids-option trustScreen]
user@host# set ip source-route-option
user@host# set ip loose-source-route-option
user@host# set ip strict-source-route-option
user@host# set ip record-route-option
```

2. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit]
user@host# set security zones security-zone trustZone screen trustScreen
```

Configuring Default Reject Rules

The following guidelines describe when to configure the default reject rules:

- Source address is defined on a multicast network, a loopback address, or a multicast address.
- The source or destination address of a packet is a link-local address, an address “reserved for future use” as specified in RFC 5735 for IPv4, an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6.
- An illegal or out-of-sequence TCP packet is received.

1. Before you begin, log in with your root account on a Junos OS device running Junos OS Release 22.2R1 and edit the configuration.

   NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure default reject rules:

1. Configure the security screen features and enable the IP address spoofing IDS option.

```
[edit]
user@host# set security screen ids-option trustScreen ip spoofing
```

2. Configure the security flow feature to log the dropped illegal packets.

```
[edit]
user@host# set security flow log dropped-illegal-packet
```

3. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit]
user@host# set security zones security-zone trustZone screen trustScreen
```

4. Configure the mandatory TCP reject rule.

```
[edit]
user@host# set security flow tcp-session strict-syn-check
```

CHAPTER 13

Configuring Network Attacks

Configuring IP Teardrop Attack Screen | 143

Configuring TCP Land Attack Screen | 144

Configuring ICMP Fragment Screen | 146

Configuring Ping-Of-Death Attack Screen | 148

Configuring tcp-no-flag Attack Screen | 150

Configuring TCP SYN-FIN Attack Screen | 151

Configuring TCP fin-no-ack Attack Screen | 153

Configuring UDP Bomb Attack Screen | 155

Configuring UDP CHARGEN DoS Attack Screen | 155

Configuring TCP SYN and RST Attack Screen | 157

Configuring ICMP Flood Attack Screen | 159

Configuring TCP SYN Flood Attack Screen | 161

Configuring TCP Port Scan Attack Screen | 163

Configuring UDP Port Scan Attack Screen | 165

Configuring IP Sweep Attack Screen | 167

Configuring IP Teardrop Attack Screen

This topic describes how to configure detection of an IP teardrop attack.

Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the field is the fragment offset fields, which indicates the starting position, or offset of the data contained in a fragmented packet, relative to the data of the original unfragmented packet. When the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap and the server attempting to reassemble the packet might crash.

To enable detection of a teardrop attack:

1. Configure interfaces and assign IP addresses to the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
```

```
permit
user@host# set security policies default-policy deny-all
```

4. Configure the security screen option and attach it to the **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen ip tear-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP Land Attack Screen

This topic describes how to configure detection of a TCP land attack.

Land attacks occur when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and the source IP address.

To enable detection of a TCP land attack:

1. Configure interfaces and assign IP addresses to the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp land
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring ICMP Fragment Screen

This topic describes how to configure detection of an ICMP fragment attack.

If an ICMP packet is large, then it must be fragmented. When the ICMP fragment protection screen option is enabled, the Junos OS blocks any ICMP packet that has many fragment flags set or that has an offset value indicated in the offset field.

To enable detection of an ICMP fragment IDS attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen icmp fragment
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring Ping-Of-Death Attack Screen

This topic describes how to configure detection of ping-of-death attack.

The IP datagram with the protocol field of the IP header is set to 1 (ICMP), the last fragment bit is set, and (IP offset * 8) + (IP data length) > 65535. The IP offset (which represents the starting position of this fragment in the original packet, and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.

To enable detection of a ping-of-death IDP attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
```

```
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen icmp ping-death
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring tcp-no-flag Attack Screen

This topic describes how to configure detection of a `tcp-no-flag` attack.

A TCP segment with no control flags set is an anomalous event causing various responses from the recipient. When the TCP no-flag is enabled, the device detects the TCP segment headers with no flags set, and drops all TCP packets with missing or malformed flag fields.

To enable detection of a `tcp-no-flag` option:

1. Configure interfaces and assign an IP address to the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp tcp-no-flag
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP SYN-FIN Attack Screen

This topic describes how to configure detection of a TCP SYN-FIN attack.

A TCP header with the SYN and FIN flags set is anomalous TCP behavior causing various responses from the recipient, depending on the OS. Blocking packets with SYN and FIN flags helps prevent the OS system probes.

To enable detection of TCP SYN-FIN bits:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp syn-fin
```

```
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP fin-no-ack Attack Screen

This topic describes how to configure detection of TCP `fin-no-ack` attack. A TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior.

To enable detection of FIN bits with no ACK bit IDS option:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp fin-no-ack
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring UDP Bomb Attack Screen

If the UDP length specified is less than the IP length specified then the malformed packet type is associated with a denial-of-service attempt. By default, SRX drops these packets. No configuration is required.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring UDP CHARGEN DoS Attack Screen

This topic describes how to configure protection from a UDP CHARGEN DoS attack.

NOTE: UDP packet is detected with a source port of 7 and a destination port of 19 is an attack.

To enable detection of a UDP CHARGEN DoS attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones `trustZone` and `untrustZone` and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from `untrustZone` to the `trustZone` with the Junos OS predefined application `junos-chargen`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application junos-chargen
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
deny
user@host# set security policies default-policy permit-all
```

4. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

5. To allow the packet to reach the destination, change the policy configuration from `deny` to `permit`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP SYN and RST Attack Screen

This topic describes how to configure TCP packet when the SYN and RST flags are set.

To enable detection of a TCP SYN and RST attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones `trustZone` the `untrustZone` and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
```

```
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure the IDP custom-attack signatures.

```
[edit]
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match from-zone any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match source-address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match to-zone any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match destination-
address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match application default
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match attacks custom-
attacks syn_rst
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then action no-action
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then notification log-
attacks
user@host# set security idp active-policy idpengine
user@host# set security idp custom-attack syn_rst severity info
user@host# set security idp custom-attack syn_rst attack-type signature context packet
user@host# set security idp custom-attack syn_rst attack-type signature pattern
user@host# set security idp custom-attack syn_rst attack-type signature direction any
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp tcp-
flags rst
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp tcp-
flags syn
```

4. Configure security policies from `untrustZone` to `trustZone`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit application-services idp
user@host# set security policies default-policy deny-all
```

5. Configure security `tcp-session` option in flow.

```
[edit]
user@host# set security flow tcp-session no-syn-check
user@host# set security flow tcp-session no-sequence-check
```

6. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

7. To allow the traffic to reach the destination, configure the `tcp-session` option.

```
[edit]
user@host# set security flow tcp-session relax-check
```

8. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring ICMP Flood Attack Screen

This topic describes how to configure detection of an ICMP flood attack.

An ICMP flood typically occurs when an ICMP echo request overloads the victim with many requests such that the ICMP echo request spends all its resources responding until it can no longer process valid network traffic. When enabling the ICMP flood protection feature, you can set a threshold that, once exceeded, invokes the ICMP flood attack protection feature.

To enable detection of an ICMP flood attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones `trustZone` and `untrustZone` and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from `untrustZone` to `trustZone`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to `untrustZone`.

```
[edit]
user@host# set security screen ids-option untrustScreen icmp flood
```

```
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP SYN Flood Attack Screen

This topic describes how to configure detection of a TCP SYN flood attack.

A SYN flood occurs when a host is so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

To enable detection of a TCP SYN flood attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones trustZone and untrustZone and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from untrustZone to trustZone.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to untrustZone.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp syn-flood
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring TCP Port Scan Attack Screen

This topic describes how to configure detection of a TCP port scan attack.

A port scan occurs when one source IP address sends an IP packet containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval.

To enable detection of a TCP port scan attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones trustZone and untrustZone and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from untrustZone to trustZone.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to untrustZone.

```
[edit]
user@host# set security screen ids-option untrustScreen tcp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring UDP Port Scan Attack Screen

This topic describes how to configure detection of a UDP port scan attack.

These attacks scan the target IP addresses for open, listening, or responsive services by targeting multiple protocols or ports on one or more target IP address using obvious (sequentially numbered) patterns of the target protocol or port numbers. The patterns are derived by randomizing the protocol or port numbers and randomizing the time delays between the transmissions.

To enable detection of a UDP port scan attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones `trustZone` and `untrustZone` and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
```

```
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from `untrustZone` to `trustZone`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to `untrustZone`.

```
[edit]
user@host# set security screen ids-option untrustScreen udp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

Configuring IP Sweep Attack Screen

This topic describes how to configure detection of an IP sweep attack.

An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined time interval (5000 microseconds is the default value). The purpose of this attack is to send ICMP packets—typically echo requests—to various hosts in the hope that at least one replies, thus uncovering an address to target.

To enable detection of an IP sweep attack:

1. Configure interfaces and assign an IP address to interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones `trustZone` and `untrustZone` and assign interfaces to them.

```
[edit]
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic system-services
all
user@host# set security zones security-zone untrustZone host-inbound-traffic protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from `untrustZone` to `trustZone`.

```
[edit]
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 match
```

```
application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to `untrustZone`.

```
[edit]
user@host# set security screen ids-option untrustScreen icmp ip-sweep
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
[edit]
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy policy1 then
log session-close
```

6. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

14

CHAPTER

Configuring the IDP Extended Package

IDP Extended Package Configuration Overview | 170

IDP Extended Package Configuration Overview

The Junos OS Intrusion Detection and Prevention (IDP) policy enables you to selectively enforce various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. It allows you to define policy rules to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy defines how your device handles the network traffic. It allows you to enforce various attack detection and prevention techniques on traffic traversing your network.

A policy is made up of rule bases, and each rule base contains a set of rules. You define rule parameters, such as traffic match conditions, action, and logging requirements, then add the rules to rule bases. After you create an IDP policy by adding rules in one or more rule bases, you can select that policy to be the active policy on your device.

To configure the IDP extended package (IPS-EP) perform the following steps:

1. Enable IPS in a security policy. See Configuring IDP Policy Rules and IDP Rulebases.
2. Configure IDP policy rules, IDP rule bases, and IDP rule actions. See Configuring IDP Policy Rules and IDP Rulebases .
3. Configure IDP custom signatures. SeeUnderstanding IDP Signature-Based Attacks
4. Update the IDP signature database. SeeIntrusion Detection and Prevention Feature Guide for Security Devices.

RELATED DOCUMENTATION

- Intrusion Detection and Prevention Feature Guide for Security Devices

15

CHAPTER

Configuring Cluster Mode

Understanding Cluster Mode | 172

Configuring L2 HA Link Encryption tunnel | 172

Configuring PKI Based L2HA Link Encryption | 179

Understanding Cluster Mode

The Administrator of the TOE can set up the Cluster Mode for High Availability (HA) by connecting dedicated HA control port of node0 and node1 as described in the article - Connecting SRX Series Devices to Create a Chassis Cluster.

The factory-default configuration does not include HA configuration. To enable HA, if the physical interfaces used by HA have some configurations, these configurations need to be removed. The two hosts constituting a chassis cluster must have identical configuration except for one being configured to node 0 and the other to node 1.

The TOE has a dedicated fxp0 interface for the HA management of the TOE. The interface for HA control link must be between the dedicated control port on each device. The fabric interface may be defined by the Administrator. After the cluster has been defined and set up by the Administrator, the two devices constituting a chassis cluster have identical cluster-id but difference node ID as one host most be node 0 and the other one node 1 with slot numbering offset of 3 for SRX5400, offset of 6 for SRX5600 devices and 12 for SRX5800 devices.

The node 1 renumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. The fabric interface remains Administrator-defined.

With L2 HA link encryption tunnel, any Security Sensitive Parameters (Critical Security Parameters) exchanged over the control link between the two chassis in cluster mode is protected using IPsec. Using IPsec for internal communication between nodes, the configuration information and IKE HA messages that passes through the chassis cluster link from the primary node to the secondary node is protected from active and passive eavesdropping. Without the internal IPsec key, an attacker cannot gain privilege access or observe traffic.

Configuring L2 HA Link Encryption tunnel

Physically connect the two devices and ensure that they are the same models. Connect the dedicated control ports on node 0 and node 1. Connect the user defined fabricated ports on node 0 and node 1. To configure two chassis in cluster mode, follow the below steps:

1. Zeroize both the SRX devices before you use for cluster. If the devices are already in cluster mode please make sure you disable them before the zeroize process. For information on how to disable chassis cluster, see Disabling a Chassis Cluster.

```
user@host> request vmhost zeroize
```

2. Delete the web management services.

```
user@host# delete system services web-management
```

3. Configure FIPS mode and bring up the devices in FIPS mode.

```
[edit]
     user@host# set groups global system fips level 2
    [edit]
     user@host# set groups global system root-authentication plain-textpassword
     New password: type password here
     Retype new password: retype password here
    [edit]
     user@host# commit
user@host> request vmhost reboot
```

4. Configure device 1 with standard cluster commands for operating in cluster mode as node0 with control port configuration. See Chassis Cluster Control Plane Interfaces.

```
[edit]
user@host# set groups node0 system host-name node0-host-name
user@host# set groups node0 system backup-router gateway-address
user@host# set groups node0 system backup-router destination value
user@host# set groups node0 interfaces fxp0 unit 0 family inet address node0-ip-address
user@host# set groups node1 system host-name node1-host-name
user@host# set groups node1 system backup-router gateway-address
user@host# set groups node1 system backup-router destination value
user@host# set groups node1 interfaces fxp0 unit 0 family inet address node1-ip-address
user@host# set apply-groups global
user@host# set apply-groups "$(node)"
user@host# delete apply-groups re0
user@host# set system ports console log-out-on-disconnect
user@host# set chassis cluster reth-count 5
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set groups global chassis cluster control-ports fpc 2 port 0
user@host# set groups global chassis cluster control-ports fpc 8 port 0
user@host# commit
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

5. After the device 1 is up, configure HA link encryption as shown in sample configuration below, commit and reboot. Device 1 needs to be configured with both node0 and node1 HA link encryption configuration before commit and reboot.

```
[edit]
user@host# set groups node0 security ike traceoptions file ikelog
user@host# set groups node0 security ike traceoptions file size 100m
user@host# set groups node0 security ike traceoptions flag all
user@host# set groups node0 security ike traceoptions level 15
user@host# set groups node0 security ike proposal IKE_PROP_PSK authentication-method pre-
shared-keys
user@host# set groups node0 security ike proposal IKE_PROP_PSK dh-group group20
user@host# set groups node0 security ike proposal IKE_PROP_PSK authentication-algorithm
sha-256
user@host# set groups node0 security ike proposal IKE_PROP_PSK encryption-algorithm aes-256-
cbc
user@host# set groups node0 security ike policy IKE_POL_PSK proposals IKE_PROP_PSK
user@host# prompt groups node0 security ike policy IKE_POL_PSK pre-shared-key ascii-text
New ascii-text (secret): juniper
Retype new ascii-text (secret): juniper
user@host# set groups node0 security ike gateway S2S_GW ike-policy IKE_POL_PSK
user@host# set groups node0 security ike gateway S2S_GW version v2-only
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK protocol esp
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK authentication-algorithm
hmac-sha1-96
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK encryption-algorithm
aes-256-cbc
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK lifetime-seconds 200
user@host# set groups node0 security ipsec policy IPSEC_POL_PSK perfect-forward-secrecy keys
group20
user@host# set groups node0 security ipsec policy IPSEC_POL_PSK proposal IPSEC_PROP_PSK
user@host# set groups node0 security ipsec vpn S2S_VPN ha-link-encryption
user@host# set groups node0 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node0 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PSK
user@host# set groups node1 security ike traceoptions file ikelog
user@host# set groups node1 security ike traceoptions file size 100m
user@host# set groups node1 security ike traceoptions flag all
user@host# set groups node1 security ike traceoptions level 15
user@host# set groups node1 security ike proposal IKE_PROP_PSK authentication-method pre-
shared-keys
user@host# set groups node1 security ike proposal IKE_PROP_PSK dh-group group20
user@host# set groups node1 security ike proposal IKE_PROP_PSK authentication-algorithm
```

```
sha-256
user@host# set groups node1 security ike proposal IKE_PROP_PSK encryption-algorithm aes-256-
cbc
user@host# set groups node1 security ike policy IKE_POL_PSK proposals IKE_PROP_PSK
user@host# prompt groups node1 security ike policy IKE_POL_PSK pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype new ascii-text (secret): juniper
user@host# set groups node1 security ike gateway S2S_GW ike-policy IKE_POL_PSK
user@host# set groups node1 security ike gateway S2S_GW version v2-only
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK protocol esp
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK authentication-algorithm
hmac-sha1-96
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK encryption-algorithm
aes-256-cbc
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK lifetime-seconds 200
user@host# set groups node1 security ipsec policy IPSEC_POL_PSK perfect-forward-secrecy keys
group20
user@host# set groups node1 security ipsec policy IPSEC_POL_PSK proposals IPSEC_PROP_PSK
user@host# set groups node1 security ipsec vpn S2S_VPN ha-link-encryption
user@host# set groups node1 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node1 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PSK
user@host# set groups global interfaces fab0 fabric-options member-interfaces xe-1/0/3
user@host# set groups global interfaces fab1 fabric-options member-interfaces xe-7/0/3
user@host# commit
user@host> request vmhost reboot
```

6. To proceed further with device 2 configuration and commit, you need to ensure device 1 and device 2 are not reachable to each other. One way to achieve this is to power off device 1 at this point.
7. Configure device 2 with standard cluster commands for operating in cluster mode as node1 with control port configuration. See Chassis Cluster Control Plane Interfaces.

```
[edit]
user@host# set groups node0 system host-name node0-host-name
user@host# set groups node0 system backup-router gateway-address
user@host# set groups node0 system backup-router destination value
user@host# set groups node0 interfaces fxp0 unit 0 family inet address node0-ip-address
user@host# set groups node1 system host-name node1-host-name
user@host# set groups node1 system backup-router gateway-address
user@host# set groups node1 system backup-router destination value
user@host# set groups node1 interfaces fxp0 unit 0 family inet address node1-ip-address
user@host# set apply-groups global
user@host# set apply-groups "${node}"
```

```
user@host# delete apply-groups re0
user@host# set system ports console log-out-on-disconnect
user@host# set chassis cluster reth-count 5
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set groups global chassis cluster control-ports fpc 2 port 0
user@host# set groups global chassis cluster control-ports fpc 8 port 0
user@host# commit
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

8. After the device 2 is up, configure HA link encryption as shown in sample configuration below on device 2. Device 2 needs to be configured with both node0 and node1 HA link encryption configuration. Commit on node1 (device 2), and finally reboot node1 (device 2).

```
[edit]
user@host# set groups node0 security ike traceoptionsfile ikelog
user@host# set groups node0 security ike traceoptions file size 100m
user@host# set groups node0 security ike traceoptions flag all
user@host# set groups node0 security ike traceoptions level 15
user@host# set groups node0 security ike proposal IKE_PROP_PSK authentication-method pre-
shared-keys
user@host# set groups node0 security ike proposal IKE_PROP_PSK dh-group group20
user@host# set groups node0 security ike proposal IKE_PROP_PSK authentication-algorithm
sha-256
user@host# set groups node0 security ike proposal IKE_PROP_PSK encryption-algorithm aes-256-
cbc
user@host# set groups node0 security ike policy IKE_POL_PSK proposals IKE_PROP_PSK
user@host# prompt groups node0 security ike policy IKE_POL_PSK pre-shared-key ascii-text
New ascii-text (secret): juniper
Retype new ascii-text (secret): juniper
user@host# set groups node0 security ike gateway S2S_GW ike-policy IKE_POL_PSK
user@host# set groups node0 security ike gateway S2S_GW version v2-only
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK protocol esp
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK authentication-algorithm
hmac-sha1-96
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK encryption-algorithm
aes-256-cbc
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PSK lifetime-seconds 200
user@host# set groups node0 security ipsec policy IPSEC_POL_PSK perfect-forward-secrecy keys
group20
user@host# set groups node0 security ipsec policy IPSEC_POL_PSK proposal IPSEC_PROP_PSK
user@host# set groups node0 security ipsec vpn S2S_VPN ha-link-encryption
```

```
user@host# set groups node0 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node0 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PSK
user@host# set groups node1 security ike traceoptions file ikelog
user@host# set groups node1 security ike traceoptions file size 100m
user@host# set groups node1 security ike traceoptions flag all
user@host# set groups node1 security ike traceoptions level 15
user@host# set groups node1 security ike proposal IKE_PROP_PSK authentication-method pre-
shared-keys
user@host# set groups node1 security ike proposal IKE_PROP_PSK dh-group group20
user@host# set groups node1 security ike proposal IKE_PROP_PSK authentication-algorithm
sha-256
user@host# set groups node1 security ike proposal IKE_PROP_PSK encryption-algorithm aes-256-
cbc
user@host# set groups node1 security ike policy IKE_POL_PSK proposals IKE_PROP_PSK
user@host# prompt groups node1 security ike policy IKE_POL_PSK pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype new ascii-text (secret): juniper
user@host# set groups node1 security ike gateway S2S_GW ike-policy IKE_POL_PSK
user@host# set groups node1 security ike gateway S2S_GW version v2-only
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK protocol esp
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK authentication-algorithm
hmac-sha1-96
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK encryption-algorithm
aes-256-cbc
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PSK lifetime-seconds 200
user@host# set groups node1 security ipsec policy IPSEC_POL_PSK perfect-forward-secrecy keys
group20
user@host# set groups node1 security ipsec policy IPSEC_POL_PSK proposals IPSEC_PROP_PSK
user@host# set groups node1 security ipsec vpn S2S_VPN ha-link-encryption
user@host# set groups node1 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node1 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PSK
user@host# set groups global interfaces fab0 fabric-options member-interfaces xe-1/0/3
user@host# set groups global interfaces fab1 fabric-options member-interfaces xe-7/0/3
user@host# commit
user@host> request vmhost reboot
```

NOTE: To enable HA link encryption on node1 in step 6, the other node must be in lost state for the commit to complete. Consider the timing for the other node to be in lost state, else step 6 must be redone.

Configuring PKI Based L2HA Link Encryption

- Physically connect the two devices and ensure that they are the same models.
- Connect the dedicated control ports on node 0 and node 1.
- Connect the user defined fabricated ports on node 0 and node 1.

To configure two chassis in cluster mode, follow the below steps:

1. Zeroize both the SRX devices before you use for cluster. If the devices are already in cluster mode please ensure you disable them before zeroize. For information on how to disable chassis cluster, see Disabling a Chassis Cluster.
2. Delete the web management services.

   user@host# **delete system services web-management https**
3. Configure FIPS mode and bring up the devices in FIPS mode.

```
[edit]
user@host# set groups global system fips level 2
[edit]
user@host# set groups global system root-authentication plain-textpassword
```

```
New password: type password here
Retype new password: retype password here
[edit]
user@host# commit
user@host> request vmhost reboot
```

4. Configure device 1 with standard cluster commands for operating in cluster mode as node0. This requires a reboot.

```
[edit]
user@host# set groups node0 system host-name node0-host-name
user@host# set groups node0 system backup-router gateway-address
user@host# set groups node0 system backup-router destination value
user@host# set groups node0 interfaces fxp0 unit 0 family inet address node0-ip-address
user@host# set groups node1 system host-name node1-host-name
user@host# set groups node1 system backup-router gateway-address
user@host# set groups node1 system backup-router destination value
user@host# set groups node1 interfaces fxp0 unit 0 family inet address node1-ip-address
user@host# set apply-groups global
user@host# set apply-groups "$(node)"
user@host# delete apply-groups re0
user@host# set system ports console log-out-on-disconnect
user@host# set chassis cluster reth-count 5
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set groups global chassis cluster control-ports fpc 2 port 0
user@host# set groups global chassis cluster control-ports fpc 8 port 0
user@host# commit
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

See https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-chassis-cluster-verification.html

5. After the device 1 is up, configure HA link encryption as shown in sample configuration below, commit and reboot. device 1 needs to be configured with both node0 and node1 HA link encryption configuration before commit and reboot.

```
[edit]
user@host# set groups node0 security ike traceoptions file ikelog
user@host# set groups node0 security ike traceoptions file size 100m
user@host# set groups node0 security ike traceoptions flag all
```

```
user@host# set groups node0 security ike traceoptions level 15
user@host# set groups node0 security pki traceoptions file pkilog
user@host# set groups node0 security pki traceoptions file size 100m
user@host# set groups node0 security pki traceoptions flag all
user@host# set groups node0 security ike proposal IKE_PROP_PKI authentication-method rsa-
signatures
user@host# set groups node0 security ike proposal IKE_PROP_PKI dh-group group20
user@host# set groups node0 security ike proposal IKE_PROP_PKI authentication-algorithm
sha-256
user@host# set groups node0 security ike proposal IKE_PROP_PKI encryption-algorithm aes-256-
cbc
user@host# set groups node0 security ike policy IKE_POL_PKI mode main
user@host# set groups node0 security ike policy IKE_POL_PKI proposals IKE_PROP_PKI
user@host# set groups node0 security ike policy IKE_POL_PKI certificate local-certificate
pkicert
user@host# set groups node0 security ike gateway S2S_GW ike-policy IKE_POL_PKI
user@host# set groups node0 security ike gateway S2S_GW version v2-only
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI protocol esp
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI authentication-algorithm
hmac-sha1-96
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI encryptionalgorithm
aes-128-cbc
user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI lifetime-seconds 200
user@host# set groups node0 security ipsec policy IPSEC_POL_PKI perfect-forward-secrecy
keys group20
user@host# set groups node0 security ipsec policy IPSEC_POL_PKI proposals IPSEC_PROP_PKI
user@host# set groups node0 security ipsec vpn S2S_VPN ha-link-encryption
user@host# set groups node0 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node0 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PKI
user@host# set groups node0 security pki ca-profile S2S_PKI ca-identity S2S_PKI_CA1
user@host# set groups node0 security pki ca-profile S2S_PKI enrollment url <Enrollment URL
of certificate authority>
user@host# set groups node0 security pki ca-profile S2S_PKI revocation-check crl url <CRL
distribution point for certificate authority>
user@host# set groups node0 security pki ca-profile S2S_PKI revocation-check disable
user@host# set groups node0 interfaces st0 unit 0 family inet
user@host# set groups node1 security ike traceoptions file ikelog
user@host# set groups node1 security ike traceoptions file size 100m
user@host# set groups node1 security ike traceoptions flag all
user@host# set groups node1 security ike traceoptions level 15
user@host# set groups node1 security pki traceoptions file pkilog
user@host# set groups node1 security pki traceoptions file size 100m
user@host# set groups node1 security pki traceoptions flag all
```

```
user@host# set groups node1 security ike proposal IKE_PROP_PKI authentication-method rsa-
signatures
user@host# set groups node1 security ike proposal IKE_PROP_PKI dh-group group20
user@host# set groups node1 security ike proposal IKE_PROP_PKI authentication-algorithm
sha-256
user@host# set groups node1 security ike proposal IKE_PROP_PKI encryption-algorithm aes-256-
cbc
user@host# set groups node1 security ike policy IKE_POL_PKI mode main
user@host# set groups node1 security ike policy IKE_POL_PKI proposals IKE_PROP_PKI
user@host# set groups node1 security ike policy IKE_POL_PKI certificate local-certificate
pkicert
user@host# set groups node1 security ike gateway S2S_GW ike-policy IKE_POL_PKI
user@host# set groups node1 security ike gateway S2S_GW version v2-only
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI protocol esp
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI authenticationalgorithm
hmac-sha1-96
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI encryptionalgorithm
aes-128-cbc
user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI lifetime-seconds 200
user@host# set groups node1 security ipsec policy IPSEC_POL_PKI perfect-forward-secrecy
keys group20
user@host# set groups node1 security ipsec policy IPSEC_POL_PKI proposals IPSEC_PROP_PKI
user@host# set groups node1 security ipsec vpn S2S_VPN ha-link-encryption
user@host# set groups node1 security ipsec vpn S2S_VPN ike gateway S2S_GW
user@host# set groups node1 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PKI
user@host# set groups node1 security pki ca-profile S2S_PKI ca-identity S2S_PKI_CA1
user@host# set groups node1 security pki ca-profile S2S_PKI enrollment url <Enrollment URL
of certificate authority>
user@host# set groups node1 security pki ca-profile S2S_PKI revocation-check crl url <CRL
distribution point for certificate authority>
user@host# set groups node1 security pki ca-profile S2S_PKI revocation-check disable
user@host# set groups node1 interfaces st0 unit 0 family inet
user@host# set groups global interfaces fab0 fabric-options member-interfaces xe-1/0/3
user@host# set groups global interfaces fab1 fabric-options member-interfaces xe-7/0/3
user@host# commit
user@host> clear security pki node-local local-certificate all
user@host> clear security pki node-local certificate-request all
user@host> clear security pki node-local key-pair all
user@host> clear security pki crl all
user@host> clear security pki ca-certificate all
user@host> request security pki node-local generate-key-pair certificate-id pkicert type
```

```
rsa size 2048
```

```
root@vm# curl "http://<PKI-Server-IP>/certsrv/certnew.cer?
ReqID=CACert=0=bin" -o /tmp/dut_ca.cer
root@vm# scp /tmp/dut_ca.cer root@node0-host-name:/var/tmp
user@host> request security pki ca-certificate load ca-profile S2S_PKI filename/var/tmp/
dut_ca.cer
user@host> show security pki ca-certificate
```

```
root@vm# curl "http://PKI-Server-IP/certsrv/certcrl.crl?Renewal=0=bin"
-o /tmp/dut.crl
root@vm# scp /tmp/dut.crl root@node0-host-name:/var/tmp
user@host> request security pki crl load ca-profile S2S_PKI filename /var/tmp/dut.crl
user@host> show security pki crl
user@host> request security pki node-local generate-certificate-request certificate-id
pkicert subject
 CN=testdut,OU=QA,O=JuniperNetworks,L=CNRD,ST=Beijing,C=CN domainname dut.juniper.net
 ip-address 129.16.0.1 email dut@juniper.net
```

```
root@vm# rm -rf /cert
root@vm# mkdir /cert
root@vm# chmod 777 /cert
root@vm# echo -----BEGIN CERTIFICATE REQUEST-----copy-generatedkey-----END CERTIFICATE
REQUEST----- /cert/dsakey
root@vm# cat /cert/dsakey
root@vm# chmod 777 /cert/dsakey
root@vm# chmod o+w /tftpboot
root@vm# rm -f /etc/xinetd.d/tftp.org
root@vm# cp /etc/xinetd.d/tftp /etc/xinetd.d/tftp.org
root@vm# sed -e 's/server_args.*/server_args = -s \/tftpboot -c/g' /etc/xinetd.d/tftp  /etc/
xinetd.d/tftp.mdf
root@vm# mv -f /etc/xinetd.d/tftp.mdf /etc/xinetd.d/tftp
root@vm# systemctl enable tftp.service
root@vm# /bin/systemctl restart xinetd.service
root@vm# mv -f /etc/xinetd.d/tftp.org /etc/xinetd.d/tftp
root@vm# dir /tftpboot/pki.tcl
root@vm# /bin/cp /tftpboot/pki.tcl /cert/
root@vm# chmod 775 /cert/pki.tcl
```

```
root@vm# /cert/pki.tcl PKI-Server-IP /cert/dsakey /cert/dut.cer
root@vm# scp /cert/dut.cer root@node0-host-name:/var/tmp
```

6. To proceed further with device 2 configuration and commit, you need to ensure device1 and device 2 are not reachable to each other. One way to achieve this is to power off device 1 at this point.
7. Configure device 2 with standard cluster command for operating in cluster mode as node1. This requires a reboot.

   [edit]

   user@host# set groups node0 system host-name node0-host-name

   user@host# set groups node0 system backup-router gateway-address

   user@host# set groups node0 system backup-router destination value

   user@host# set groups node0 interfaces fxp0 unit 0 family inet address node0-ip-address

   user@host# set groups node1 system host-name node1-host-name

   user@host# set groups node1 system backup-router gateway-address

   user@host# set groups node1 system backup-router destination value

   user@host# set groups node1 interfaces fxp0 unit 0 family inet address node1-ip-address

   user@host# set apply-groups global

   user@host# set apply-groups “$(node)”

   user@host# delete apply-groups re0

   user@host# set system ports console log-out-on-disconnect

   user@host# set chassis cluster reth-count 5

   user@host# set chassis cluster redundancy-group 0 node 0 priority 254

   user@host# set chassis cluster redundancy-group 0 node 1 priority 1

   user@host# set groups global chassis cluster control-ports fpc 2 port 0

   user@host# set groups global chassis cluster control-ports fpc 8 port 0

   user@host# commit

   user@host> set chassis cluster cluster-id 1 node 1 reboot

   See https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-chassis-cluster-verification.html

8. After the device 2 is up, configure HA link encryption as shown in sample configuration below on device 2. Device 2 needs to be configured with both node0 and node1 HA link encryption configuration. Commit on node1 (device 2), and finally reboot node1 (device 2).

   [edit]

   user@host# set groups node0 security ike traceoptions file ikelog

   user@host# set groups node0 security ike traceoptions file size 100m

   user@host# set groups node0 security ike traceoptions flag all

   user@host# set groups node0 security ike traceoptions level 15

   user@host# set groups node0 security pki traceoptions file pkilog

   user@host# set groups node0 security pki traceoptions file size 100m

   user@host# set groups node0 security pki traceoptions flag all

   user@host# set groups node0 security ike proposal IKE_PROP_PKI authentication-method

   rsa-signatures

   user@host# set groups node0 security ike proposal IKE_PROP_PKI dh-group group20

   user@host# set groups node0 security ike proposal IKE_PROP_PKI authentication-algorithm sha-256

   user@host# set groups node0 security ike proposal IKE_PROP_PKI encryption-algorithm aes-256-cbc

   user@host# set groups node0 security ike policy IKE_POL_PKI mode main

   user@host# set groups node0 security ike policy IKE_POL_PKI proposals IKE_PROP_PKI

   user@host# set groups node0 security ike policy IKE_POL_PKI certificate local-certificate pkicert

   user@host# set groups node0 security ike gateway S2S_GW ike-policy IKE_POL_PKI

   user@host# set groups node0 security ike gateway S2S_GW version v2-only

   user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI protocol esp

   user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI authenticationalgorithm

   hmac-sha1-96

   user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI encryptionalgorithm

   aes-128-cbc

   user@host# set groups node0 security ipsec proposal IPSEC_PROP_PKI lifetime-seconds 200

user@host# set groups node0 security ipsec policy IPSEC_POL_PKI perfect-forwardsecrecy keys group20

user@host# set groups node0 security ipsec policy IPSEC_POL_PKI proposals IPSEC_PROP_PKI

user@host# set groups node0 security ipsec vpn S2S_VPN ha-link-encryption

user@host# set groups node0 security ipsec vpn S2S_VPN ike gateway S2S_GW

user@host# set groups node0 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PKI

user@host# set groups node0 security pki ca-profile S2S_PKI ca-identity S2S_PKI_CA1

user@host# set groups node0 security pki ca-profile S2S_PKI enrollment url <Enrollment URL of certificate authority>

user@host# set groups node0 security pki ca-profile S2S_PKI revocation-check crl url <CRL distribution point for certificate authority>

user@host# set groups node0 security pki ca-profile S2S_PKI revocation-check disable

user@host# set groups node0 interfaces st0 unit 0 family inet

user@host# set groups node1 security ike traceoptions file ikelog

user@host# set groups node1 security ike traceoptions file size 100m

user@host# set groups node1 security ike traceoptions flag all

user@host# set groups node1 security ike traceoptions level 15

user@host# set groups node1 security pki traceoptions file pkilog

user@host# set groups node1 security pki traceoptions file size 100m

user@host# set groups node1 security pki traceoptions flag all

user@host# set groups node1 security ike proposal IKE_PROP_PKI authentication-method rsa-signatures

user@host# set groups node1 security ike proposal IKE_PROP_PKI dh-group group20

user@host# set groups node1 security ike proposal IKE_PROP_PKI authentication-algorithm sha-256

user@host# set groups node1 security ike proposal IKE_PROP_PKI encryption-algorithm aes-256-cbc

user@host# set groups node1 security ike policy IKE_POL_PKI mode main

user@host# set groups node1 security ike policy IKE_POL_PKI proposals IKE_PROP_PKI

user@host# set groups node1 security ike policy IKE_POL_PKI certificate local-certificate pkicert

user@host# set groups node1 security ike gateway S2S_GW ike-policy IKE_POL_PKI

user@host# set groups node1 security ike gateway S2S_GW version v2-only

user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI protocol esp

user@host#set groups node1 security ipsec proposal IPSEC_PROP_PKI authenticationalgorithm hmac-sha1-96

user@host> set groups node1 security ipsec proposal IPSEC_PROP_PKI encryptionalgorithm aes-128-cbc

user@host# set groups node1 security ipsec proposal IPSEC_PROP_PKI lifetime-seconds 200

user@host# set groups node1 security ipsec policy IPSEC_POL_PKI perfect-forward-secrecy keys group20

user@host# set groups node1 security ipsec policy IPSEC_POL_PKI proposals IPSEC_PROP_PKI

user@host# set groups node1 security ipsec vpn S2S_VPN ha-link-encryption

user@host# set groups node1 security ipsec vpn S2S_VPN ike gateway S2S_GW

user@host# set groups node1 security ipsec vpn S2S_VPN ike ipsec-policy IPSEC_POL_PKI

user@host# set groups node1 security pki ca-profile S2S_PKI ca-identity S2S_PKI_CA1

user@host# set groups node1 security pki ca-profile S2S_PKI enrollment url <Enrollment URL of certificate authority>

user@host# set groups node1 security pki ca-profile S2S_PKI revocation-check crl url <CRL distribution point for certificate authority>

user@host# set groups node1 security pki ca-profile S2S_PKI revocation-check disable

user@host# set groups node1 interfaces st0 unit 0 family inet

user@host# set groups global interfaces fab0 fabric-options member-interfaces xe-1/0/3

user@host# set groups global interfaces fab1 fabric-options member-interfaces xe-7/0/3

user@host# commit

user@host> clear security pki node-local local-certificate all

user@host> clear security pki node-local certificate-request all

user@host> clear security pki node-local key-pair all

user@host> clear security pki crl all

user@host> clear security pki ca-certificate all

user@host> request security pki node-local generate-key-pair certificate-id pkicert type rsa size 2048

```
root@vm# curl "http://PKI-Server-IP/certsrv/certnew.cer?
ReqID=CACert=0=bin" -o /tmp/aux_ca.cer
root@vm# scp /tmp/aux_ca.cer root@node1-host-name:/var/tmp
```

user@host> request security pki ca-certificate load ca-profile S2S_PKI filename/var/tmp/aux_ca.cer

user@host> show security pki ca-certificate

```
root@vm# curl "http://PKI-Server-IP/certsrv/certcrl.crl?Renewal=0=bin"
-o /tmp/aux.crl
root@vm# scp /tmp/aux.crl root@node1-host-name:/var/tmp
```

user@host> request security pki crl load ca-profile S2S_PKI filename /var/tmp/aux.crl

user@host> show security pki crl

user@host> request security pki node-local generate-certificate-request certificate-id pkicert subject

CN=testaux,OU=QA,O=JuniperNetworks,L=CNRD,ST=Beijing,C=CN domainname aux.juniper.net

ip-address 130.16.0.1 email aux@juniper.net

```
root@vm# rm -rf /cert
root@vm# mkdir /cert
root@vm# chmod 777 /cert
root@vm# echo -----BEGIN CERTIFICATE REQUEST-----copy-generatedkey-----
END CERTIFICATE REQUEST-----  /cert/dsakey
root@vm# cat /cert/dsakey
root@vm# chmod 777 /cert/dsakey
root@vm# chmod o+w /tftpboot
root@vm# rm -f /etc/xinetd.d/tftp.org
root@vm# cp /etc/xinetd.d/tftp /etc/xinetd.d/tftp.org
```

```
root@vm# sed -e 's/server_args.*/server_args = -s \/tftpboot -c/g' /etc/
xinetd.d/tftp  /etc/xinetd.d/tftp.mdf
root@vm# mv -f /etc/xinetd.d/tftp.mdf /etc/xinetd.d/tftp
root@vm# systemctl enable tftp.service
root@vm# /bin/systemctl restart xinetd.service
root@vm# mv -f /etc/xinetd.d/tftp.org /etc/xinetd.d/tftp
root@vm# dir /tftpboot/pki.tcl
root@vm# /bin/cp /tftpboot/pki.tcl /cert/
root@vm# chmod 775 /cert/pki.tcl
root@vm# /cert/pki.tcl PKI-Server-IP /cert/dsakey /cert/aux.cer
root@vm# scp /cert/aux.cer root@node1-host-name:/var/tmp
```

```
user@host> clear security pki node-local local-certificate all
         user@host> request security pki node-local local-certificate load filename
           /var/tmp/aux.cer
         certificate-id pkicert
         user@host> request vmhost reboot
```

9. Power ON node0 (device 1).
10. Both the nodes will be in cluster mode with HA link encryption enabled.

NOTE: To enable HA link encryption on node1 in step 6, the other node must be in lost state for the commit to go through. Hence, manage the timing correctly, else step 6 must be redone until enabling HA link encryption on node1 commit goes through. The above example shows, configuring PKI based L2 HA link encryption tunnel with RSA. However, we can also use ECDSA with key size 256 and 384.

16
CHAPTER

Performing Self-Tests on a Device

Understanding FIPS Self-Tests | 191

Understanding FIPS Self-Tests

IN THIS SECTION

- Performing Power-On Self-Tests on the Device | **192**

The cryptographic module enforces security rules to ensure that a device running the Juniper Networks Junos operating system (Junos OS) in FIPS mode of operation meets the security requirements of FIPS 140-3 Level 2. To validate the output of cryptographic algorithms approved for FIPS and test the integrity of some system modules, the device performs the following series of known answer test (KAT) self-tests:

- `kernel_kats`—KAT for kernel cryptographic routines
- `md_kats`—KAT for libmd and libc
- `openssl_kats`—KAT for OpenSSL cryptographic implementation
- `openssl-102_kats`—KAT for OpenSSL v1.0.2 cryptographic implementation
- `quicksec_7_0_kats`—KAT for Quicksec_7_0Toolkit cryptographic implementation
- `srxpfe_kats`—KAT for SRX packet forwarding engine

The KAT self-tests are performed automatically at startup and reboot, when FIPS mode of operation is enabled on the device. Conditional self-tests are also performed automatically to verify digitally signed software packages, generated random numbers, RSA and ECDSA key pairs, and manually entered keys.

If the KATs are completed successfully, the system log (syslog) file is updated to display the tests that were executed.

If the device fails a KAT, the device writes the details to a system log file, enters FIPS error state (panic), and reboot.

The `file show /var/log/messages` command displays the system log.

Proceed with normal operation after the reboot is complete. If an error occurs, please contact the Juniper Networks Technical Assistance Center (JTAC).

You must have administrative privileges to configure FIPS self-tests. The device must be running the evaluated version of Junos OS in FIPS mode software.

In this example, the FIPS self-test is executed at 9:00 AM in New York City, USA, every Wednesday.

NOTE: Instead of weekly tests, you can configure monthly tests by including the month and day-of-month statements.

Performing Power-On Self-Tests on the Device

Each time the cryptographic module is powered on, the module tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged. Power-on self-tests are performed on demand by power cycling the module. On powering on or resetting the device, the module performs the following self-tests. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fail, the module enters the Critical Failure error state. The module displays the following status output for SRX5400 and SRX5800 devices while running the power-on self-tests:

```
Initializing Verified Exec:
        random: randomdev_wait_until_seeded unblock wait
        uhub0: 21 ports with 21 removable, self powered
        random: Entropy start-up health tests performed on 1024 samples passed.
        random: unblocking device.
        FIPS veriexec ECDSA Verify Known Answer Test: Passed
        Verified os-kernel-prd-x86-64-20220607 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Enforcing Verified Exec:
        Verified os-libs-12-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Mounting os-libs-12-x86-64-20220607.2c547a1_builder_stable_12_222
        Verified os-runtime-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Mounting os-runtime-x86-64-20220607.2c547a1_builder_stable_12_222
        ** /dev/gpt/config
        FILE SYSTEM CLEAN; SKIPPING CHECKS
        clean, 426502 free (6 frags, 53312 blocks, 0.0% fragmentation)
        ** /dev/gpt/var
        FILE SYSTEM CLEAN; SKIPPING CHECKS
        clean, 12942661 free (317 frags, 1617793 blocks, 0.0% fragmentation)
        @ 1663137800 [2022-09-14 06:43:20 UTC] verify active ...
```

```
        Verified jail-runtime-x86-32-20220607 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified fips-optest-x86-32-22.9 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jdocs-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified dsa-x86-64-22.9 signed by PackageProductionECP256_2022 method ECDSA256+SHA256
        Verified fips-mode-x86-64-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jinsight-x86-32-22.9 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jpfe-common-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256

        Verified jpfe-X960-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jpfe-X-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jmrt-base-x86-64-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jfirmware-x86-32-22.8 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jpfe-spc3-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jpfe-wrlinuxlts19-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-appid-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-aacl-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-alg-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jsd-x86-32-22.9-jet-1 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jservices-cos-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-cpcd-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-crypto-base-x86-32-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified jservices-hcm-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
```

```
        Verified jservices-idp-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-dnsf-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-ids-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-ipsec-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-jflow-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-llpdf-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-lrf-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-jdpi-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-mobile-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-mss-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256

        Verified jservices-nat-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-pcef-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-rpm-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-rtcom-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-sfw-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-softwire-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-tcp-log-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-telemetry-x86-32-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified jservices-traffic-dird-x86-32-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified jservices-ssl-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
```

```
        Verified junos-daemons-srx-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified jservices-urlf-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-daemons-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-dp-crypto-support-srx-x86-32-20220617 signed by
PackageProductionECP256_2022
        method ECDSA256+SHA256
        Verified junos-appsecure-he-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-ike-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified junos-l2-rsi-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified junos-libs-compat32-srx-x86-64-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified junos-libs-srx-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-modules-srx-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-libs-compat32-x86-64-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified junos-libs-x86-64-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified junos-modules-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256

        Verified junos-probe-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-net-mtx-prd-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-platform-srx-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-openconfig-x86-32-22.9 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-platform-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-routing-compat32-x86-64-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
```

```
        Verified junos-redis-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-routing-aggregated-x86-64-20220617 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified junos-routing-lsys-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-runtime-srx-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified junos-routing-mpls-oam-basic-x86-64-20220617 signed by
PackageProductionECP256_2022
        method ECDSA256+SHA256
        Verified junos-runtime-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified na-telemetry-x86-32-22.9 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified jweb-srx-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified junos-net-prd-x86-64-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified os-boot-junos-ve-x86-64-20220607 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified os-compat32-x86-64-20220607 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified os-libs-12-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified os-kernel-prd-x86-64-20220607 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256
        Verified os-crypto-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified os-runtime-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified os-vmguest-x86-64-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified os-libs-compat32-12-x86-64-20220607 signed by PackageProductionECP256_2022
method
        ECDSA256+SHA256
        Verified py-base-x86-32-20220617 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        Verified py-extensions-x86-32-20220617 signed by PackageProductionECP256_2022 method
        ECDSA256+SHA256


        Verified junos-vmguest-mtx-x86-64-20220617 signed by PackageProductionECP256_2022 method
```

```
        ECDSA256+SHA256
        Verified os-zoneinfo-20220607 signed by PackageProductionECP256_2022 method
ECDSA256+SHA256
        @ 1663137801 [2022-09-14 06:43:21 UTC] verify done
        @ 1663137801 [2022-09-14 06:43:21 UTC] mount start
        @ 1663137801 [2022-09-14 06:43:21 UTC] junos 22.2R1.9
        Mounting os-zoneinfo-20220607.2c547a1_builder_stable_12_222
        Mounting junos-net-prd-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-libs-x86-64-20220617.153850_builder_junos_222_r1
        Mounting os-libs-compat32-12-x86-64-20220607.2c547a1_builder_stable_12_222
        Mounting os-compat32-x86-64-20220607.2c547a1_builder_stable_12_222
        Mounting junos-libs-compat32-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-runtime-x86-32-20220617.153850_builder_junos_222_r1
        Starting watchdog daemon ...
        Mounting junos-vmguest-mtx-x86-64-20220617.153850_builder_junos_222_r1
        Mounting py-extensions-x86-32-20220617.153850_builder_junos_222_r1
        Mounting py-base-x86-32-20220617.153850_builder_junos_222_r1
        Mounting os-vmguest-x86-64-20220607.2c547a1_builder_stable_12_222
        Mounting os-crypto-x86-64-20220607.2c547a1_builder_stable_12_222
        Mounting na-telemetry-x86-32-22.2R1.9
        Mounting junos-libs-compat32-srx-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-runtime-srx-x86-32-20220617.153850_builder_junos_222_r1
        Mounting junos-routing-mpls-oam-basic-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-routing-lsys-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-routing-compat32-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-routing-aggregated-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-redis-x86-32-20220617.153850_builder_junos_222_r1
        Mounting junos-probe-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-platform-x86-32-20220617.153850_builder_junos_222_r1
        Mounting junos-platform-srx-x86-32-20220617.153850_builder_junos_222_r1
        Mounting junos-openconfig-x86-32-22.2R1.9
        Mounting junos-modules-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-modules-srx-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-libs-srx-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-l2-rsi-20220617.153850_builder_junos_222_r1
        Mounting junos-dp-crypto-support-srx-x86-32-20220617.153850_builder_junos_222_r1
        Mounting junos-daemons-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-daemons-srx-x86-64-20220617.153850_builder_junos_222_r1
        Mounting junos-appsecure-he-x86-32-20220617.153850_builder_junos_222_r1
        Mounting jsd-x86-32-22.2R1.9-jet-1
        Mounting jpfe-wrlinuxlts19-x86-32-20220617.153850_builder_junos_222_r1
        Mounting jpfe-spc3-x86-32-20220617.153850_builder_junos_222_r1
        Mounting jpfe-X960-x86-32-20220617.153850_builder_junos_222_r1
```

```
        Mounting jpfe-common-x86-32-20220617.153850_builder_junos_222_r1
        Mounting jpfe-X-x86-32-20220617.153850_builder_junos_222_r1
        Mounting jmrt-base-x86-64-20220617.153850_builder_junos_222_r1
        Mounting jinsight-x86-32-22.2R1.9
        Mounting jfirmware-x86-32-22.2R1.8
        Mounting jdocs-x86-32-20220617.153850_builder_junos_222_r1
        Mounting fips-optest-x86-32-22.2R1.9
        Mounting fips-mode-x86-64-20220617.153850_builder_junos_222_r1
        Mounting dsa-x86-64-22.2R1.9
        @ 1663137842 [2022-09-14 06:44:02 UTC] mount done
        grep: /var/etc/jlaunchd.inc: No such file or directory
        grep: /var/etc/jlaunchd.inc: No such file or directory
        grep: /var/etc/jlaunchd.inc: No such file or directory
        grep: /var/etc/jlaunchd.inc: No such file or directory
        Removing /etc/malloc.conf
        Checking platform support for: srx5400
        @ 1663137844 [2022-09-14 06:44:04 UTC] mountlate start
        Mounting jweb-srx-x86-32-20220617.153850_builder_junos_222_r1
        Setup /packages/mnt/jweb-srx-5d585241/jail/var/cache dir only for srx5400
        mount_nullfs: /web-api: No such file or directory
        Mounting junos-ike-x86-32-20220617.153850_builder_junos_222_r1
        @ 1663137848 [2022-09-14 06:44:08 UTC] mountlate done
        kern.module_path: /packages/sets/active/boot/os-vmguest/;/packages/sets/active/boot/
netstack/;/
        packages/sets/active/boot/os-crypto/;/packages/sets/active/boot/os-kernel/;/packages/
sets/active/
        boot/junos-net-platform/;/packages/sets/active/boot/junos-modules/ -> /modules;/modules/
dev;/
        modules/ifpfe_drv;/modules/ifpfe_media;/modules/jam_core;/modules/jam_plugin;/modules/
peertype;/
        modules/platform
        besw0: mem 0xfeb80000-0xfeb8ffff irq 10 at device 5.0 on pci0
        Loading BCMSDK module.....
        bcm_sdk_init(): DevID = 0xb680, RevID = 0x12
        bcm_sdk_init: device ID: dev: 0xb680, rev: 0x12
        bcm_sdk_init: device unit no: 0
        bcm_soc_cm_device_init: device unit no: 0
        bcore_init: after soc_reset_init
        bcore_init: after soc_misc_init
        bcore_init: after soc_mmu_init
        bcore_init: before bcm_init
        bcore_init: before port stuff
```

```
        bcore_init: after port stuff
        bcore_init: link scan interval is (soc_property): 4000000
        bcore_mxseries_init: Finished mxseries port configuration
        bcore_init: Finished platform specific initialization

        bcm_sdk_init: Done sdk init
        Loading JUNOS chassis module
        chassis_init_hw_chassis_startup_time: chassis startup time 0.000000, shared:
0x7fffffff300,
        base: 0x7fffffff000, offset: 0x300
        IPsec: Initialized Security Association Processing.
        hgcommdev0: port 0xc000-0xc0ff mem 0xfeba8000-0xfeba8fff at device 22.0 on pci0
        hgcommdev0: hgcommdev: registers at 0xfffff800feba8000
        pci-hgcomdev module loadedLoading the CHMIC module
        Loading POS driver
        Loading Aggregate sonet driver
        Loading the SLB driver
        Loading the IMA Group Media Layer; Attaching to media services layer
        Loading the IMA Link Media Layer; Attaching to media services layer
        Loading the SONET Media Layer; Attaching to media services layer
        Loading the Protobuf-C module
        Loading the JAM-Core module
        Loading the JAM-Core module - succeeded
        Loading Multilink Services PICs module.
        Loading the Mx Platform NETPFE module
        MTX Platform JAM-Core module - load success
        interface pci_hgcommdev.1 already present in the KLD 'pci-hgcomm.ko'!
        linker_load_file: /modules/platform/pci_hgcomm.ko - unsupported file type
        kldload: an error occurred while loading module pci_hgcomm.ko. Please check dmesg(8) for
more
        details.
        Junosprocfs mounted on /junosproc.
        VirtIO PCI 9P Transport adapter is not present
        @ 1663137852 [2022-09-14 06:44:12 UTC] mgd start
        Creating initial configuration: ...
        mgd: Running FIPS Self-tests
        mgd: Testing kernel KATS:
        mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
        mgd: DES3-CBC Known Answer Test: Passed
        mgd: HMAC-SHA1 Known Answer Test: Passed
        mgd: HMAC-SHA2-256 Known Answer Test: Passed
        mgd: SHA-2-384 Known Answer Test: Passed
        mgd: SHA-2-512 Known Answer Test: Passed
```

```
mgd: AES128-CMAC Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
mgd: Testing MACSec KATS:
mgd: AES128-CMAC Known Answer Test: Passed
mgd: AES256-CMAC Known Answer Test: Passed
mgd: AES-ECB Known Answer Test: Passed

mgd: AES-KEYWRAP Known Answer Test: Passed
mgd: KBKDF Known Answer Test: Passed
mgd: Testing libmd KATS:
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: SHA-2-512 Known Answer Test: Passed
mgd: Testing OpenSSL v1.0.2 KATS:
mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
mgd: FIPS ECDSA Known Answer Test: Passed
mgd: FIPS ECDH Known Answer Test: Passed
mgd: FIPS RSA Known Answer Test: Passed
mgd: DES3-CBC Known Answer Test: Passed
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-224 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: HMAC-SHA2-384 Known Answer Test: Passed
mgd: HMAC-SHA2-512 Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
mgd: AES-GCM Known Answer Test: Passed
mgd: ECDSA-SIGN Known Answer Test: Passed
mgd: KDF-IKE-V1 Known Answer Test: Passed
mgd: KDF-SSH-SHA256 Known Answer Test: Passed
mgd: KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed
mgd: KAS-FFC-EPHEM-NOKC Known Answer Test: Passed
mgd: Testing OpenSSL KATS:
mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
mgd: FIPS ECDSA Known Answer Test: Passed
mgd: FIPS ECDH Known Answer Test: Passed
mgd: FIPS RSA Known Answer Test: Passed
mgd: DES3-CBC Known Answer Test: Passed
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-224 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: HMAC-SHA2-384 Known Answer Test: Passed
mgd: HMAC-SHA2-512 Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
```

```
mgd: AES-GCM Known Answer Test: Passed
mgd: ECDSA-SIGN Known Answer Test: Passed
mgd: KDF-IKE-V1 Known Answer Test: Passed
mgd: KDF-SSH-SHA256 Known Answer Test: Passed
mgd: KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed
mgd: KAS-FFC-EPHEM-NOKC Known Answer Test: Passed
mgd: Testing QuickSec 7.0 KATS:

mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
mgd: DES3-CBC Known Answer Test: Passed
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-224 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: HMAC-SHA2-384 Known Answer Test: Passed
mgd: HMAC-SHA2-512 Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
mgd: AES-GCM Known Answer Test: Passed
mgd: SSH-RSA-ENC Known Answer Test: Passed
mgd: SSH-RSA-SIGN Known Answer Test: Passed
mgd: SSH-ECDSA-SIGN Known Answer Test: Passed
mgd: KDF-IKE-V1 Known Answer Test: Passed
mgd: KDF-IKE-V2 Known Answer Test: Passed
mgd: Testing QuickSec KATS:
mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
mgd: DES3-CBC Known Answer Test: Passed
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-224 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: HMAC-SHA2-384 Known Answer Test: Passed
mgd: HMAC-SHA2-512 Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
mgd: AES-GCM Known Answer Test: Passed
mgd: SSH-RSA-ENC Known Answer Test: Passed
mgd: SSH-RSA-SIGN Known Answer Test: Passed
mgd: KDF-IKE-V1 Known Answer Test: Passed
mgd: KDF-IKE-V2 Known Answer Test: Passed
mgd: Testing SSH IPsec KATS:
mgd: NIST 800-90 HMAC DRBG Known Answer Test: Passed
mgd: DES3-CBC Known Answer Test: Passed
mgd: HMAC-SHA1 Known Answer Test: Passed
mgd: HMAC-SHA2-256 Known Answer Test: Passed
mgd: AES-CBC Known Answer Test: Passed
mgd: SSH-RSA-ENC Known Answer Test: Passed
```

```
        mgd: SSH-RSA-SIGN Known Answer Test: Passed
        mgd: KDF-IKE-V1 Known Answer Test: Passed
        mgd: Testing file integrity:
        mgd: File integrity Known Answer Test: Passed
        mgd: Testing crypto integrity:
        mgd: Crypto integrity Known Answer Test: Passed
        mgd: Expect an exec Authentication error...
        MAC/veriexec: no fingerprint (file=/sbin/kats/cannot-exec fsid=225 fileid=49356 gen=1
uid=0
        206
        pid=8369 ppid=8335 gppid=8333)mgd: /sbin/kats/run-tests: /sbin/kats/cannot-exec:
Authentication
        error
        mgd: FIPS Self-tests Passed
```

NOTE: The module implements cryptographic libraries and algorithms that are not utilized in the approved mode of operation.

The module displays the following status output for SRX5400 and SRX5800 devices while failure of the power-on self-tests:

```
 Testing kernel KATS:
      panic: pid 2121 (kernel_kats), uid 0, FIPS error 1: NIST 800-90 HMAC DRBG Known Answer
Test:
      Failed
      Testing libmd KATS:
panic: pid 91115 (md_kats), uid 0, FIPS error 1: HMAC-SHA1 Known Answer Test: Failed
Testing OpenSSL v1.0.2 KATS:
panic: pid 20121 (openssl-102_kats), uid 0, FIPS error 1: NIST 800-90 HMAC DRBG Known Answer Test:
Failed
      Testing OpenSSL KATS:
      panic: pid 2340 (openssl_kats), uid 0, FIPS error 1: NIST 800-90 HMAC DRBG Known Answer
Test:
      Failed
      Testing QuickSec 7.0 KATS:
      panic: pid 37538 (quicksec_7_0_kats), uid 0, FIPS error 1: NIST 800-90 HMAC DRBG Known
Answer
      Test: Failed
```

17

CHAPTER

Configuration Statements

checksum-validate | 204

code | 206

data-length | 207

destination-option | 209

extension-header | 211

header-type | 212

home-address | 214

identification | 216

icmpv6 (Security IDP Custom Attack) | 218

ihl (Security IDP Custom Attack) | 220

option-type | 221

reserved (Security IDP Custom Attack) | 223

routing-header | 225

sequence-number (Security IDP ICMPv6 Headers) | 226

type (Security IDP ICMPv6 Headers) | 228

checksum-validate

IN THIS SECTION

- Syntax | 204
- Hierarchy Level | 204
- Description | 205
- Options | 205
- Required Privilege Level | 205
- Release Information | 205

Syntax

```
checksum-validate {
    match (equal | greater-than | less-than | not-equal);
    value checksum-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol ipv4]
[edit security idp custom-attack attack-name attack-type signature protocol tcp]
[edit security idp custom-attack attack-name attack-type signature protocol udp]
[edit security idp custom-attack attack-name attack-type signature protocol icmp]
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Description

Allow IDP to validate checksum field against the calculated checksum.

Options

match (equal | greater-than | less-than | not-equal)

Match an operand.

value *checksum-value*

Match a decimal value.

- **Range:** 0 through 65,535

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

code

IN THIS SECTION

- Syntax | 206
- Hierarchy Level | 206
- Description | 206
- Options | 207
- Required Privilege Level | 207
- Release Information | 207

Syntax

```
code {
    match (equal | greater-than | less-than | not-equal);
    value code-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Description

Specify the secondary code that identifies the function of the request/reply within a given type.

Options

- `match` (`equal` | `greater-than` | `less-than` | `not-equal`)—Match an operand.
- `value` *`code-value`*—Match a decimal value.
- **Range:** 0 through 255

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

data-length

IN THIS SECTION

- Syntax | 208
- Hierarchy Level | 208
- Description | 208
- Options | 208
- Required Privilege Level | 209

Release Information | 209

Syntax

```
data-length {
    match (equal | greater-than | less-than | not-equal);
    value data-length;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol udp]
[edit security idp custom-attack attack-name attack-type signature protocol icmp]
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
[edit security idp custom-attack attack-name attack-type signature protocol tcp]
```

Description

Specify the number of bytes in the data payload. In the TCP header, for SYN, ACK, and FIN packets, this field should be empty.

Options

- `match` (`equal` | `greater-than` | `less-than` | `not-equal`)—Match an operand.
- `value` *`data-length`*—Match the number of bytes in the data payload.
- **Range:** 0 through 65,535

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 9.3.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

destination-option

IN THIS SECTION

- Syntax | 209
- Hierarchy Level | 210
- Description | 210
- Required Privilege Level | 210
- Release Information | 210

Syntax

```
destination-option {
    home-address {
        match (equal | greater-than | less-than | not-equal);
        value header-value;
```

```
    }
    option-type {
        match (equal | greater-than | less-than | not-equal);
        value header-value;
    }
}
```

Hierarchy Level

```
[edit  set security idp custom-attack attack-name attack-type signature protocol ipv6 extension-
header]
```

Description

Specify the IPv6 destination option for the extension header. The `destination-option` option inspects the header option type of `home-address` field in the `extension header` and reports a custom attack if a match is found. The `destination-option` supports the `home-address` field type of inspection.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

extension-header

IN THIS SECTION

- Syntax | 211
- Hierarchy Level | 212
- Description | 212
- Required Privilege Level | 212
- Release Information | 212

Syntax

```
extension-header {
    destination-option {
        home-address {
            match (equal | greater-than | less-than | not-equal);
            value header-value;
        }
            option-type {
                match (equal | greater-than | less-than | not-equal);
                value header-value;
            }
    }
    routing-header {
        header-type {
            match (equal | greater-than | less-than | not-equal);
            value header-value;
            }
        }
}
```

Hierarchy Level

```
[edit  set security idp custom-attack attack-name attack-type signature protocol ipv6]
```

Description

Specify the IPv6 extension header.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

header-type

IN THIS SECTION

- Syntax | 213
- Hierarchy Level | 213

- Description | 213
- Options | 213
- Required Privilege Level | 214
- Release Information | 214

Syntax

```
header-type {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
}
```

Hierarchy Level

```
[edit  set security idp custom-attack attack-name attack-type signature protocol ipv6 extension-header routing-header]
```

Description

Specify the IPv6 routing header type.

Options

match (equal | greater-than | less-than | not-equal)

Match an operand.

value

Match a decimal value.

- **Range:** 0 through 255

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

home-address

IN THIS SECTION

- Syntax | 214
- Hierarchy Level | 215
- Description | 215
- Options | 215
- Required Privilege Level | 215
- Release Information | 215

Syntax

```
home-address {
    match (equal | greater-than | less-than | not-equal);
```

```
    value value;
}
```

Hierarchy Level

```
[edit  set security idp custom-attack attack-name attack-type signature protocol ipv6 extension-
header destination-option]
```

Description

Specify the IPv6 home address of the mobile node.

Options

match (equal | greater-than | less-than | not-equal)	Match an operand.
value	Match a decimal value.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

identification

IN THIS SECTION

- Syntax | 216
- Hierarchy Level | 216
- Description | 217
- Options | 217
- Required Privilege Level | 217
- Release Information | 217

Syntax

```
identification {
    match (equal | greater-than | less-than | not-equal);
    value identification-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Description

Specify a unique value used by the destination system to associate requests and replies.

Options

- `match` (`equal` | `greater-than` | `less-than` | `not-equal`)—Match an operand.
- `value` *`identification-value`*—Match a decimal value.
- **Range:** 0 through 65,535

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | **170**

icmpv6 (Security IDP Custom Attack)

IN THIS SECTION

- Syntax | 218
- Hierarchy Level | 219
- Description | 219
- Required Privilege Level | 219
- Release Information | 219

Syntax

```
icmpv6 {
    checksum-validate {
        match (equal | greater-than | less-than | not-equal);
        value checksum-value;
    }
    code {
        match (equal | greater-than | less-than | not-equal);
        value code-value;
    }
    data-length {
        match (equal | greater-than | less-than | not-equal);
        value data-length;
    }
    identification {
        match (equal | greater-than | less-than | not-equal);
        value identification-value;
    }
    sequence-number {
        match (equal | greater-than | less-than | not-equal);
        value sequence-number;
    }
    type {
        match (equal | greater-than | less-than | not-equal);
```

```
        value type-value;
    }
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol]
```

Description

Allow IDP to match the attack for the specified ICMPv6.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

ihl (Security IDP Custom Attack)

IN THIS SECTION

- Syntax | 220
- Hierarchy Level | 220
- Description | 220
- Options | 221
- Required Privilege Level | 221
- Release Information | 221

Syntax

```
ihl {
    match (equal | greater-than | less-than | not-equal);
    value ihl-value;
}
```

Hierarchy Level

```
[edit  set security idp custom-attack ipv4_custom attack-type signature protocol ipv4]
```

Description

Specify the IPv4 header length in words.

Options

match (equal | greater-than | less-than | not-equal)

Match an operand.

value

Match a decimal value.

- **Range:** 0 through 15

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

option-type

IN THIS SECTION

- Syntax | 222
- Hierarchy Level | 222
- Description | 222
- Options | 222
- Required Privilege Level | 223

Release Information | 223

Syntax

```
option-type {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol ipv6 extension-
header destination-option]
```

Description

Specify the type of option for destination header type.

Options

match (equal | greater-than | less-than | not-equal)

Match an operand.

value

Match a decimal value.

- **Range:** 0 through 255

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

reserved (Security IDP Custom Attack)

IN THIS SECTION

- Syntax | 223
- Hierarchy Level | 224
- Description | 224
- Options | 224
- Required Privilege Level | 224
- Release Information | 224

Syntax

```
reserved {
    match (equal | greater-than | less-than | not-equal);
```

```
    value reserved-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack ipv4_custom attack-type signature protocol tcp]
```

Description

Specify the three reserved bits in the TCP header field.

Options

match (equal | greater-than | less-than | not-equal)

Match an operand.

value

Match a decimal value.

- **Range:** 0 through 7

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

routing-header

IN THIS SECTION

- Syntax | 225
- Hierarchy Level | 225
- Description | 226
- Required Privilege Level | 226
- Release Information | 226

Syntax

```
routing-header {
    header-type {
        match (equal | greater-than | less-than | not-equal);
        value header-value;
    }
}
```

Hierarchy Level

```
[edit  set security idp custom-attack attack-name attack-type signature protocol ipv6 extension-
header]
```

Description

Specify the IPv6 routing header type. The `routing-header` option inspects the routing-header type field and reports a custom attack if a match with the specified value is found. The `routing-header` option supports the following routing header types: `routing-header-type0`, `routing-header-type1`, and so on.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170

sequence-number (Security IDP ICMPv6 Headers)

IN THIS SECTION

- Syntax | 227
- Hierarchy Level | 227
- Description | 227
- Options | 227
- Required Privilege Level | 227
- Release Information | 228

Syntax

```
sequence-number {
    match (equal | greater-than | less-than | not-equal);
    value sequence-number;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Description

Specify the sequence number of the packet. This number identifies the location of the request/reply in relation to the entire sequence.

Options

- `match` (`equal` | `greater-than` | `less-than` | `not-equal`)—Match an operand.
- `value` *`sequence-number`*—Match a decimal value.
- **Range:** 0 through 65,535

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview

type (Security IDP ICMPv6 Headers)

IN THIS SECTION

- Syntax | 228
- Hierarchy Level | 229
- Description | 229
- Options | 229
- Required Privilege Level | 229
- Release Information | 229

Syntax

```
type {
    match (equal | greater-than | less-than | not-equal);
    value type-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Description

Specify the primary code that identifies the function of the request/reply.

Options

`match` (`equal` | `greater-than` | `less-than` | `not-equal`)—Match an operand.

`value` *`type-value`*—Match a decimal value.

- **Range:** 0 through 255

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview | 170