CONFIGURATION GUIDE

RUCKUS FastIron FIPS and Common Criteria Configuration Guide, 09.0.10

Supporting FastIron Software Release 09.0.10c

Part Number: 53-100573-01
Publication Date: 16 June 2022

Copyright, Trademark and Proprietary Rights Information

Export Restrictions

These products and associated technical data (in print or electronic form) may be subject to export control laws of the United States of America. It is your responsibility to determine the applicable regulations and to comply with them. The following notice is applicable for all products or technology subject to export control:

These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. government or as otherwise authorized by U.S. law and regulations.

Disclaimer

THIS CONTENT AND ASSOCIATED PRODUCTS OR SERVICES ("MATERIALS"), ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT PERMISSIBLE PURSUANT TO APPLICABLE LAW, COMMSCOPE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, FREEDOM FROM COMPUTER VIRUS, AND WARRANTIES ARISING FROM COURSE OF DEALING OR COURSE OF PERFORMANCE. CommScope does not represent or warrant that the functions described or contained in the Materials will be uninterrupted or error-free, that defects will be corrected, or are free of viruses or other harmful components. CommScope does not make any warranties or representations regarding the use of the Materials in terms of their completeness, correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise. As a condition of your use of the Materials, you warrant to CommScope that you will not make use thereof for any purpose that is unlawful or prohibited by their associated terms of use.

Limitation of Liability

IN NO EVENT SHALL COMMSCOPE, COMMSCOPE AFFILIATES, OR THEIR OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, SUPPLIERS, LICENSORS AND THIRD PARTY PARTNERS, BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER, EVEN IF COMMSCOPE HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT, OR ANY OTHER THEORY ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIALS. Because some jurisdictions do not allow limitations on how long an implied warranty lasts, or the exclusion or limitation of liability for consequential or incidental damages, some of the above limitations may not apply to you.

Trademarks

ARRIS, the ARRIS logo, COMMSCOPE, RUCKUS, RUCKUS WIRELESS, the Ruckus logo, the Big Dog design, BEAMFLEX, CHANNELFLY, FASTIRON, ICX, SMARTCELL and UNLEASHED are trademarks of CommScope, Inc. and/or its affiliates. Wi-Fi Alliance, Wi-Fi, the Wi-Fi logo, Wi-Fi Certified, the Wi-Fi CERTIFIED logo, Wi-Fi Protected Access, the Wi-Fi Protected Setup logo, Wi-Fi Protected Setup, Wi-Fi Multimedia and WPA2 and WMM are trademarks or registered trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

Contents

Preface

Contacting RUCKUS Customer Services and Support

The Customer Services and Support (CSS) organization is available to provide assistance to customers with active warranties on their RUCKUS products, and customers and partners with active support contracts.

For product support information and details on contacting the Support Team, go directly to the RUCKUS Support Portal using https://support.ruckuswireless.com, or go to https://www.commscope.com/ruckus and select **Support**.

What Support Do I Need?

Technical issues are usually described in terms of priority (or severity). To determine if you need to call and open a case or access the self-service resources, use the following criteria:

- Priority 1 (P1)—Critical. Network or service is down and business is impacted. No known workaround. Go to the **Open a Case** section.
- Priority 2 (P2)—High. Network or service is impacted, but not down. Business impact may be high. Workaround may be available. Go to the **Open a Case** section.
- Priority 3 (P3)—Medium. Network or service is moderately impacted, but most business remains functional. Go to the **Self-Service Resources** section.
- Priority 4 (P4)—Low. Requests for information, product documentation, or product enhancements. Go to the **Self-Service Resources** section.

Open a Case

When your entire network is down (P1), or severely impacted (P2), call the appropriate telephone number listed below to get help:

- Continental United States: 1-855-782-5871
- Canada: 1-855-782-5871
- Europe, Middle East, Africa, Central and South America, and Asia Pacific, toll-free numbers are available at https://support.ruckuswireless.com/contact-us and Live Chat is also available.
- Worldwide toll number for our support organization. Phone charges will apply: +1-650-265-0903

We suggest that you keep a physical note of the appropriate support number in case you have an entire network outage.

Self-Service Resources

The RUCKUS Support Portal at https://support.ruckuswireless.com offers a number of tools to help you to research and resolve problems with your RUCKUS products, including:

- Technical Documentation—https://support.ruckuswireless.com/documents
- Community Forums—https://forums.ruckuswireless.com/
- Knowledge Base Articles—https://support.ruckuswireless.com/answers
- Software Downloads and Release Notes—https://support.ruckuswireless.com/#products_grid
- Security Bulletins—https://support.ruckuswireless.com/security

Using these resources will help you to resolve some issues, and will provide TAC with additional data from your troubleshooting analysis if you still require assistance through a support case or RMA. If you still require help, open and manage your case at https://support.ruckuswireless.com/case_management.

Document Feedback

RUCKUS is interested in improving its documentation and welcomes your comments and suggestions.

You can email your comments to RUCKUS at #Ruckus-Docs@commscope.com.

When contacting us, include the following information:

- Document title and release number
- Document part number (on the cover page)
- Page number (if appropriate)

For example:

- RUCKUS SmartZone Upgrade Guide, Release 5.0
- Part number: 800-71850-001 Rev A
- Page 7

RUCKUS Product Documentation Resources

Visit the RUCKUS website to locate related documentation for your product and additional RUCKUS resources.

Release Notes and other user documentation are available at https://support.ruckuswireless.com/documents. You can locate the documentation by product or perform a text search. Access to Release Notes requires an active support contract and a RUCKUS Support Portal user account. Other technical documentation content is available without logging in to the RUCKUS Support Portal.

White papers, data sheets, and other product documentation are available at https://www.commscope.com/ruckus .

Online Training Resources

To access a variety of online RUCKUS training modules, including free introductory courses to wireless networking essentials, site surveys, and products, visit the RUCKUS Training Portal at https://commscopeuniversity.myabsorb.com/. The registration is a two-step process described in this video. You create a CommScope account and then register for, and request access for, CommScope University.

Document Conventions

The following table lists the text conventions that are used throughout this guide.

TABLE 1 Text Conventions

Convention	Description	Example
`monospace`	Identifies command syntax examples	`device(config)# interface ethernet 1/1/6`
bold	User interface (UI) components such as screen or page names, keyboard keys, software buttons, and field names	On the **Start** menu, click **All Programs**.
italics	Publication titles	Refer to the *RUCKUS Small Cell Release Notes* for more information.

Notes, Cautions, and Safety Warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A NOTE provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An ATTENTION statement indicates some information that you must read before continuing with the current action or task.

CAUTION
A CAUTION statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A DANGER statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Command Syntax Conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{**x**| **y**| **z**}	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x|**y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

About This Document

Supported hardware and software

- To determine if the RUCKUS device and current software version are FIPS-certified, refer to http://csrc.nist.gov/groups/STM/cmvp/validation.html.
- To determine if the RUCKUS device and current software version is Common Criteria certified, refer to https://www.niap-ccevs.org/Product/.

Federal Information Processing Standards

FIPS Overview

A RUCKUS device in Federal Information Processing Standards (FIPS) mode is compliant with the standards established by the United States government and the National Institute of Standards and Technology (NIST).

> **NOTE**
> Not all software releases support FIPS. Refer to the Release notes for the software you are running to see if it supports FIPS.

The FIPS Publication 140-3 is a technical standard and worldwide de-facto standard for the implementation of cryptographic modules. The FIPS Publication 140-3 contains security standards developed by the United States government and the National Institute of Standards and Technology (NIST) for use by all non-military government agencies and by government contractors. Due to their importance within the security industry, these standards form a baseline for many security requirements.

In FIPS mode, the network processing occurs in the kernel and in privileged daemons.

> **NOTE**
> To determine if the device and current software version are FIPS-certified, refer to http://csrc.nist.gov/groups/STM/cmvp/validation.html.

You can configure the device to run in FIPS mode to ensure that the device is operating according to the standards stated in FIPS Publication 140-3.

A device is FIPS 140-3-compliant when the following requirements have been met:

- The device software is placed in FIPS mode with the FIPS security policy applied.

For a list of physical ports and interfaces for each device, refer to the Security Policy.

How FIPS Works

You place a device in FIPS mode by entering the **fips enable** CLI command on the management station while the station is connected to the device console port with a serial cable. After you enter the **fips enable** command, the device is administratively in FIPS mode and by default runs in strict FIPS-compliant mode upon reload.

The default FIPS policy is for the system to run in a strict mode that fully supports FIPS 140-3 specifications. However, the device allows you the flexibility to configure a modified FIPS policy according to your network requirements.

> **NOTE**
> A FIPS policy that varies from the default policy weakens the intent of the FIPS 140-3 specifications; when implemented, the device is not operating in full compliance with these specifications. Refer to Modifying the FIPS Policy on page 36.

The default FIPS policy enforces the following actions for strict FIPS compliance:

- Disables TFTP access
- Disables monitor access to memory access commands
- Returns 0 or null for SNMP MIBs for passwords or keys (referred to as critical security parameter objects)
- Zeroizes shared secrets and passwords

The device performs the following functions automatically during reboot after the **fips enable** command is entered:

- Disables Telnet
- Enables SCP access
- Disables the HTTP server
- Disables SNMP access to critical security parameter (CSP) MIB objects

After defining the FIPS policy, save the configuration, and reboot the device. While the device is booting, several tests are run to ensure the device is FIPS-compliant.

After these tests are completed successfully, the device reloads and is operationally in FIPS mode.

All the optional FIPS policy commands are provided to perform various non-approved FIPS operations when FIPS is enabled.

NOTE
If any of these policy commands are configured, the module is not operating in the approved FIPS mode.

NOTE
HTTPS server is enabled in FIPS/CC mode. HTTP server is disabled in FIPS/CC mode.

Upgrading and Downgrading Software on FIPS-enabled Devices

Software Downgrades from FIPS

Once FIPS mode has been enabled on an ICX device, downgrading to an earlier FastIron software version is not recommended. Downgrading an ICX device, even if done after disabling FIPS mode, could result in the device continuously rebooting or becoming otherwise unusable.

Upgrading FIPS-enabled Devices

FIPS 140-3 compliance is a combination of implemented hardware procedures and the activation of a software-based security policy.

FIPS 140-3 certification is achieved when the device meets certain physical security and software conditions:

- FIPS software compliance: The devices are configured to run in FIPS operational mode with the default FIPS security policy.

  NOTE
  Although commands to alter the FIPS security policy exist, altering the default FIPS security policy is not recommended.

  NOTE
  After enabling FIPS mode on your device, you cannot disable it without losing the device configuration. To disable FIPS mode, it is recommended that you contact RUCKUS Technical Support and perform the procedure under qualified guidance.

Preparing for a FIPS Software Upgrade

Before upgrading the software on the device, consider the following important information and notes.

FastIron devices can store two Full Layer 3 images or two Layer 2 images.

NOTE
Signature file auto-copy is not supported. Only image files will be updated and the FIPS check will fail if the corresponding signature file is not present in the same partition.

NOTE
Manual update of firmware is not supported.

Image Verification in FIPS or CC Mode

For a FIPS- or CC-enabled device running FastIron 08.0.10 or later, uploading a flash or boot code triggers a FIPS Load Qualification test that performs a digital signature verification of the flash or boot code using a signature file. In earlier FastIron versions, the test is triggered only when the FIPS- or CC-enabled device boots.

- FIPS devices running FastIron 08.0.10 or later support the digital signature files generated using the SHA-256/RSA-2048 algorithm.

- FastIron versions earlier than 08.0.10 in FIPS or CC mode support the digital signature files generated using the SHA-1/DSA-1024 algorithm.

Verifying the Currently Active Software Version

Use the **show version** command to check the active software version on a FastIron device. The following example displays the current software version of an ICX 7450 as version 09.0.10c and provides additional details on the image file and the modules installed in the device.

```
ICX7450-48F Router# show version
  Copyright (c) Ruckus Networks, Inc. All rights reserved.
    UNIT 1: compiled on Mar  3 2022 at 17:24:40 labeled as SPR09010c
      (33554432 bytes) from Primary SPR09010c.bin (UFI)
        SW: Version 09.0.10cT213
      Compressed Primary Boot Code size = 786944, Version:10.1.23T215 (spz10123)
       Compiled on Thu Jan 27 08:44:33 2022

  HW: Stackable ICX7450-48F
  Internal USB: Serial #: 9900614053000004
      Vendor: ATP Electronics, Total size = 1919 MB
==========================================================================
UNIT 1: SL 1: ICX7450-48F 48-port Management Module
      Serial  #:CYS3324K00T
      Software Package: ICX7450_L3_SOFT_PACKAGE
      Current License: l3-prem-macsec
      P-ASIC  0: type B548, rev 01  Chip BCM56548_A0
==========================================================================
UNIT 1: SL 2: ICX7400-SERVICE-MOD Module
      Serial  #:EAT3343L015
      FW Version  #:2.09
==========================================================================
UNIT 1: SL 3: ICX7400-1X40GQ 1-port 40G Module
      Serial  #:CYX3348K02H
==========================================================================
UNIT 1: SL 4: ICX7400-1X40GQ 1-port 40G Module
      Serial  #:CYX3349K0A2
==========================================================================
 1000 MHz ARM processor ARMv7 88 MHz bus
    8 MB boot flash memory
    2 GB code flash memory
    2 GB DRAM
STACKID 1  system uptime is 12 day(s) 21 hour(s) 23 minute(s) 51 second(s)
The system started at 23:49:04 GMT+00 Fri Mar 04 2022

The system : started=warm start   reloaded=by "reload"
```

Checking the Inactive Software Version in Secondary Storage

Use the **show flash** command to verify the version of the inactive image loaded in secondary flash. The show flash command displays the image version for both primary and secondary flash partitions as shown in the following example.

```
Device# show flash
Stack unit 1:
  Compressed Pri Code size = 63863960, Version:09.0.10c (TNR09010c.bin)
  Compressed Sec Code size = 63865696, Version:09.0.10c (TNR09010c.bin)
  Compressed Pri Boot Code size = 1573376, Version:10.1.18T235 (tnu10118)
  Compressed Sec Boot Code size = 1573376, Version:10.1.18T235 (tnu10118)
  Code Flash Free Space = 2637774848
```

Performing a FIPS or CC Software Upgrade to FastIron 09.0.10c

To upgrade the FastIron software image to FastIron 09.0.10c in support of a FIPS or CC environment, perform the following steps.

1. Place the new flash signature file and the new flash image in an SCP client directory to which the FastIron device has access.

2. If the ICX device is FIPS- or CC-enabled, copy the SHA-256/RSA-2048 signature file from the SCP client into flash memory as shown in the following example.

   NOTE
   If the ICX device is not FIPS- or CC-enabled, refer to FIPS Configuration on page 21 to enable FIPS or CC mode.

   NOTE
   In FIPS mode, SSH and SCP use diffie-hellman-group14-sha256 by default for Key Exchange. The SCP client used should be able to support this option. Any client with OpenSSH 7.2 or higher supports this option as does Putty 0.67 or higher.

```
ICX7450# copy scp flash 1.1.1.1 SPR09010cufi.sig fips-ufi-primary-sig
ICX7450# copy scp flash 1.1.1.1 SPR09010cufi.sig fips-ufi-secondary-sig
```

   Syntax: copy scp flash *source-ip-address signaturefile***ufi.sig** { **fips-ufi-primary-sig** | **fips-ufi-secondary-sig** }

3. Copy the image file from an scp client into flash memory as shown in the following example.

```
ICX7450# copy scp flash 1.1.1.1 SPR09010cufi.bin primary
ICX7450# copy scp flash 1.1.1.1 SPR09010cufi.bin secondary
```

   Syntax: copy scp flash *source-ip-address image-name***ufi.bin** { **primary** | **secondary** }

4. Verify that the flash code has been successfully copied by examining the console log or entering the **show flash** command at any level of the CLI.

```
--FIPS: secondary image verification success
```

   NOTE
   If image verification fails, the binary image is note saved.

5. Save the running configuration by entering the **write memory** command.
6. Reload the configuration to run the FIPS-enabled or CC-enabled image by entering the **reload** command.

   NOTE
   The encrypted device will not pass traffic during a reboot.

SSH Connection after Upgrading a FIPS or CC Operational Device to FastIron 08.0.10 or Later

If you are using key authentication for SSH connection, you should re-import the new format public keys into the device after upgrading a device in FIPS or CC operational mode.

For details on how to import public keys, refer to the section Usernames and SSH Public Key Authentication on page 24.

Downgrading from FIPS to Non-FIPS Mode

Once the ICX device is enabled for FIPS, it remains enabled internally, so that the signature file must be copied first whenever the image is copied.

Once you have downgraded to non-fips mode, you must still load the relevant signature (.sig) file before the image (.bin) file every time you upgrade or downgrade an image in the future. This prevents the ICX device from looping.

Downgrading from FIPS mode to non-FIPS mode clears all shared secrets, host passwords, SSH and HTTPS host keys and HTTPS certificates.

NOTE
Before upgrading or downgrading a major software version, zeroize the keys by executing the **crypto key zeroize** command.

NOTE
Once FIPS mode is enabled on the system, even if the mode is disabled later, a firmware integrity test will always be carried out on the device when the image is copied.

The steps to place a device in non-FIPS mode can be summarized as follows.

Downgrade to non-FIPS Mode Using the Existing Image:

1. Zeroize all keys using the **crypto key zeroize** command.
2. Disable FIPS with the **no fips enable** command.
3. Save the configuration with the **write memory** command.
4. Reload the configuration with the **reload** command.

Downgrade to non-FIPS Mode with SCP Using a New Image

1. While in FIPS mode, copy the signature file and image file using SCP.
2. Zeroize all keys using the **crypto key zeroize** command.
3. Disable FIPS with the **no fips enable** command.
4. Save the configuration with the **write memory** command.
5. Reload the configuration with the **reload** command.

Downgrade to non-FIPS Mode with FTP Using a New Image

1. Zeroize all keys using the **crypto key zeroize** command.
2. Disable FIPS with the **no fips enable** command.
3. Enable TFTP with the **no tftp disable** command.
4. Copy the signature file and image file using TFTP.

   The following example uses TFTP to copy the FastIron 09.0.10b UFI signature file to the primary and secondary flash of an ICX 7450 device.

   ```
   ICX7450# copy tftp flash 1.1.1.1 SPR09010bufi.sig fips-ufi-primary-sig
   ICX7450# copy tftp flash 1.1.1.1 SPR09010bufi.sig fips-ufi-secondary-sig
   ```

   The following example uses TFTP to copy the FastIron 09.0.10b UFI image file to the primary and secondary flash of an ICX 7450 device.

   ```
   ICX7450# copy tftp flash 1.1.1.1 SPR09010bufi.bin primary
   ICX7450# copy tftp flash 1.1.1.1 SPR09010bufi.bin secondary
   ```

5. Save the configuration with the **write memory** command.
6. Reload the configuration with the **reload** command.

The following task uses SCP to downgrade from FIPS to non-FIPS using a new image.

1. Log in to the device by entering your user name and password.

2. While still in FIPS mode, copy the desired application image and signature file with SCP.

   The following example uses SCP to copy the FastIron 09.0.10b UFI signature file to the primary and secondary flash of an ICX 7450 device.

   ```
   ICX7450# copy scp flash 1.1.1.1 SPR09010bufi.sig fips-ufi-primary-sig
   ICX7450# copy scp flash 1.1.1.1 SPR09010bufi.sig fips-ufi-secondary-sig
   ```

   The following example uses SCP to copy the FastIron 09.0.10b UFI image file to the primary and secondary flash of an ICX 7450 device.

   ```
   ICX7450# copy scp flash 1.1.1.1 SPR09010bufi.bin primary
   ICX7450# copy scp flash 1.1.1.1 SPR09010bufi.bin secondary
   ```

   Syntax: copy scp flash *source-ip-address signaturefile***ufi.sig** { **fips-ufi-primary-sig** | **fips-ufi-secondary-sig** }

   Syntax: copy scp flash *source-ip-address image-name***ufi.bin** [**primary** | **secondary**]

3. Zeroize all the keys by executing **crypto key zeroize** command.

   ```
   device# configure terminal
   device(config)# crypto key zeroize
   ```

4. Disable FIPS by entering the **no fips enable** or **no fips enable common-criteria** command at the prompt.

   ```
   device(config)# no fips enable
   device(config)# exit
   ```

5. Enter the **write memory** command to save the changes.

   ```
   device# write memory
   ```

6. Reload the configuration by entering the **reload** command.

   ```
   device# reload
   ```

Once the switch is rebooted, refer to Placing the Device in FIPS Mode on page 29 if you want to re-enable FIPS.

FIPS Configuration

User Roles in FIPS Mode

Configuring FIPS mode on the RUCKUS devices complies with the standards established by the United States government and the National Institute of Standards and Technology (NIST).

A RUCKUS device in FIPS mode supports three user roles:

- Crypto-officer role: The Crypto-officer role on the device in FIPS mode is equivalent to the administrator role, or the super-user role in non-FIPS mode.
- Port Configuration Administrator role: The Port Configuration Administrator on the device in FIPS mode is equivalent to the port configuration user in non-FIPS mode and has write access to the interface configuration mode only.
- User role: The User role on the device in FIPS mode has read-only privileges and no configuration mode access.

Concurrent operators are supported, but no limit is enforced. The number of concurrent users is only limited by the system resources.

Commands Disabled in FIPS Mode

The device in FIPS mode does not support the following commands:

- **telnet server**
- **ip ssh key-authentication no**
- **ip ssh scp disable**
- **web-management http**
- **ip ssh encryption disable-aes-cbc**

A device in FIPS mode does not support TFTP commands, including **copy tftp flash** *ip-address*.

The following JITC command is not supported because JITC is disabled by default in FIPS mode:

- **jitc enable**

Hidden Files in FIPS Mode

Hidden files are not displayed when the device is in FIPS mode. Hidden files are displayed only when the device is in non-FIPS mode.

Cryptographic Algorithms in FIPS Mode

The device in FIPS mode supports the following FIPS 140-3-approved cryptographic algorithms:

- Advanced Encryption Algorithm (AES)
- Secure Hash Algorithm (SHA) (including variants the module supports: SHA-256, SHA-384, and SHA-512)
- Keyed-Hash Message Authentication Code (HMAC)
- Deterministic Random Bit Generator (DRBG)
- Rivest, Shamir, and Adleman public key encryption algorithm (RSA)
- Elliptic curve Digital Signature Algorithm (ECDSA)
- Key-Based Key Derivation Function (KBKDF)
- KAS-FFC-SSC
- KAS-ECC-SSC
- SP800-135 KDF - TLS, SSH, IKEv2, and SNMP

Allowed exceptions include:

- --

The device in FIPS mode does not support the following cryptographic algorithms:

- Message Digest 5 (MD5)
- Hash Message Authentication Codes - Message Digest 5 (HMAC-MD5) as used in RADIUS
- Non-Deterministic Random Number Generator (NDRNG)
- RC4
- 3-DES

SSH

To enable SSH, you must generate RSA encryption keys using the following command.

```
device(config)# crypto key generate rsa  modulus 2048
device(config)#
Creating RSA key pair, please wait...
RSA Key pair is successfully created
```

To confirm that SSH is enabled, use the **show ip ssh** command.

```
device# show ip ssh
No SSH sessions are currently established
SSH-v2.0 enabled; hostkey: RSA(2048)
```

NOTE

If you zeroize the RSA keys, the SSH server is disabled.

Zeroize the keys as shown in the following example.

```
device(config)# crypto key zeroize  rsa
RSA Key pair is successfully deleted
```

Use the **show ip ssh** command to confirm that SSH is disabled.

```
device(config)# show ip ssh
No SSH sessions are currently established
SSH-v2.0 disabled
```

To establish a connection from the client side, enable a local user and set a user password. The following rules apply to passwords:

1. Passwords must be at least eight characters long.

   NOTE
   The password can be from 8 through 60 characters in length.

2. Passwords must consist of characters from three or more of these character classes: uppercase, lowercase, numerical, ASCII non-alphanumeric.
3. The password must not end with the only numeric character in the password.

NOTE
In CC mode, after three incorrect attempts to enter the password, user access is disabled until login recovery time is reached. The default recovery time is three minutes. The recovery time is configurable using the **enable user disable-on-login-failure** command.

To enable a user, enter commands similar to the following.

```
device# configure terminal
device(config)# user test password tesT123$$
```

The example enables the user "test". It then sets the user password to "tesT123$$".

NOTE
The user configured in the previous example, "test," is a crypto officer who will be able to modify or delete the configuration. To create a read-only user, use a command similar to the following example thta includes the keywords **privilege 5**.

```
device(config)# user test privilege 5 password tesT123$$
```

Login

When a user tries to log in with correct username and password, the login is successful. The following syslog message is generated for a successful login attempt:

```
SYSLOG: <14> Dec 18 09:03:26 Device Security: SSH login by admin from src IP 15.15.15.1 from src MAC
0200.8801.8132 to USER EXEC mode using RSA as Server Host Key.
```

Logout

When the user "admin" closes the session from the TOE, the session is disconnected. The following syslog message is generated for a successful logout attempt:

```
SYSLOG: <14> Dec 18 09:09:11 Device Security: SSH logout by admin from src IP 15.15.15.1 from src MAC
0200.8801.8132 from USER EXEC mode using RSA as Server Host Key.
```

Failed login attempts

By default, the user is disabled after three failed attempts to login. Each time a user connects with a wrong password, a syslog message is displayed. The following example indicates three unsuccessful login attempts.

```
SYSLOG: <14> Dec 18 09:18:14 Device Security: SSH access by user admin from src IP 15.15.15.1 rejected, 1
attempt(s)

SYSLOG: <14> Dec 18 09:18:15 Device Security: SSH access by user admin from src IP 15.15.15.1 rejected, 2
attempt(s)

SYSLOG: <14> Dec 18 09:18:16 Device Security: SSH access by user admin from src IP 15.15.15.1 rejected, 3
attempt(s)
```

The number of attempts and the time for which the user is disabled is configurable. Use the **enable user disable-on-login-failure** command with appropriate parameters to configure the number of login attempts before a user is disabled and the amount of time the system is blocked before the user is allowed to attempt login again.

The following example allows four failed login attempts before the user is disabled and the recovery time of five seconds begins.

```
device# configure terminal
device(config)# enable user disable-on-login-failure 4 login-recovery-time in-secs 5
```

Syntax:[no] enable user { **disable-on-login-failure** [*invalid-attempts* **login-recovery-time** { **in-hours** | **in-mins** | **in-secs** } *recovery-time*] }

NOTE
By default, the user is allowed three login attempts. In CC mode, the default recovery time for re-enabling user accounts is three minutes.

Login attempts can be any decimal value from 1 through 10.

Login recovery time can be specified in hours, minutes, or seconds.

NOTE
You must configure users before enabling the aaa console; otherwise, you may be logged off and locked out of the system.

SSH Clients

SSH clients must be FIPS 186-3-compliant. You can use an SSH client that is equivalent to OpenSSH v7.5 or later.

Usernames and SSH Public Key Authentication

The device stores or uses the username that is provided by the SSH client when public-key authentication is used. Therefore, the username is mentioned in the login and logout syslogs.

The FastIron devices save the username from the public-key authentication request. The username is used in the login and logout syslogs. When FIPS mode is operational, the FastIron device uses the username to match against the username attached to the SSH client public key stored on the device. If the two usernames do not match, the authentication request is denied.

Implementation

The client public key file format allows for a username to be provided in the "Subject" field the SSH2 public key. Additional private headers can be used. There are three privilege levels: 0 READ-WRITE/ADMINISTRATOR, 4 PORT-CONFIG, and 5 READ-ONLY. The following public key example shows the two headers that are used by the device. No continuation lines are allowed in the file for these headers.

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20121206"
Subject: brcd
```

```
x-device-privilege-level: 0
AAAAB3NzaC1yc2EAAAABJQAAAQEAkwiApY1x4T/DHII5JzR2OgqcF5vjlubNcvSE
UjkGmiRBDSOicjxS0ZLm1b2xFpVzw8XxSSy8cxvntfs5ortOt80QzynqgL+H2zJa
Lb4Qbu6/1vakJbPb/VUJE66Zezh0c8mze6zTbiP4iQ/Wn2lxpSmlS5cdowmFlZ7B
97xcagJIBl+7JKuvj8P+85ESUf2/pcrogqx7gdr1IpP2nev5s4xwCWFGtr2R/yMF
Q9h0xLcc4A7vLTDuY/h1GzLdICgtNYdqpUhpw+w0DkTKbQuDPd0gkwHkoFwg85lE
4VCDevdC/DeOCNJjNp9NbVD+SW6uL4NymmV7/i0YbPyl3gTESQ==
---- END SSH2 PUBLIC KEY ----
```

After decoding the base64 encoded public keys to binary format, a SHA256 hash of the binary format key is created. This hash is saved to memory. The hash is verified as unique compared to all the hashes of client public keys that have already been parsed. Non-empty usernames are also verified as unique compared to the usernames already parsed in the public key. Access is denied if the usernames are mismatched.

The username has the following restrictions:

- The username cannot contain control characters, spaces, ", ?, |, or characters above ASCII code 0x7F.
- The username must be less than or equal to 48 characters.
- The username must be specified with the public key for that key to allow access. The user must specify a non-empty username in the login request.

Restrictions

- No EXEC authorization from AAA: No EXEC authorization through the AAA server is available because the privilege level is obtained from the public key file private header field (x-device-privilege-level) as shown in the public key example in Implementation on page 24.
- No EXEC accounting from AAA
- No system accounting from AAA

Protocol Changes in FIPS Mode

The following table lists the protocols that undergo changes while the device is in FIPS mode with the default policy applied.

TABLE 2 Protocol Changes

Protocols / Algorithms	Supported in FIPS Mode	Supported in Non-FIPS Mode	For more information on individual protocol changes, refer to the following sections:
BGP	Yes	Yes	BGP on page 26
HTTP	No	Yes	HTTP on page 26
HTTPS	Yes	Yes	HTTPS on page 26
NTP	No	Yes (supports SHA1 and MD5 hash only)	Configuring NTP on page 47
PKI	Yes	Yes	PKI on page 26
Proprietary 2-way encryption algorithms	No	Yes	Proprietary 2-way Encryption Algorithms on page 27
RADIUS	Yes	Yes	RADIUS Protocol in FIPS Mode on page 27
SCP	Yes	Yes	SCP on page 27
SSHv2	Yes, with limitations	Yes	SSHv2 on page 27
Telnet	No	Yes	Telnet on page 28
TFTP	No	Yes	TFTP on page 28

BGP

Border Gateway Protocol (BGP) allows peer-to-peer authentication or client-to-server authentication.

To authorize an authentication, use a command such as the following to configure shared secret keys for BGP:

```
device(config-bgp-router)# neighbor 192.168.1.2 password P@$$w0rd
```

Syntax: [no] neighbor {*ip-addr* | *peer-group-name*} **password** *string*

HTTP

HTTP is not supported on FastIron devices in FIPS mode.

HTTPS

The following HTTPS configurations are affected in FIPS mode:

The FIPS 140-3 cipher suites consist of the following algorithms:

- AES (FIPS 197) for symmetric key encryption and decryption.
- Secure Hash Standard (SHA-256, SHA-384, and SHA-512) (FIPS 180-2) for hashing).
- HMAC (FIPS 198) for keyed hash
- Random number generator Hash DRBG (NIST SP800-90).
- Diffie-Hellman, EC Diffie-Hellman key exchange
- RSA (PKCS #1 v2.1) for signature generation and verification for all ICX platforms; for ICX 7450 platform, RSA (PKCS #1 v2.1) or ECDSA (ANSI X9.62)

The following cipher suites are allowed in FIPS mode:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

TLS Implementation in FastIron Devices

TLS versions 1.1 and 1.2 are supported on devices that act as an HTTPS server.

For devices that act as an SSL server or HTTPS server, the default connection is with TLS 1.2. For devices that act as an SSL client or syslog, OpenFlow, or secure AAA client, during session negotiation, the TLS version is decided based on the server support.

Use the **show ip ssl** command to identify the TLS version that is configured on the device.

```
device(config)# show ip ssl
Session Protocol  Source IP      Source Port  Remote IP     Remote Port
1       TLS_1_2   10.20.157.102  634          10.25.105.201 60892
```

PKI

Public Key Infrastructure (PKI) operates on the module that allows automated certificate authentication.

The following parameters make up the PKI critical security parameters (CSPs):

- PKI SCEP Enrollment RSA 2048-bit Private Key
- PKI SCEP Enrollment RSA 2048-bit Public Key

NOTE
OCSP does not use an RSA private key.

Proprietary 2-way Encryption Algorithms

The routing protocols OSPFv2 and BGP, and the management protocol SNMP save authentications parameters using one of the following two proprietary algorithms:

- Global encoding scheme
- Base 64 encoding scheme

These proprietary algorithms are not supported in FIPS mode. When the default FIPS policy is applied, these authentication parameters are zeroized.

RADIUS Protocol in FIPS Mode

HMAC-MD5 authentication used in RADIUS is allowed in FIPS mode.

RADIUS allows client-to-server authentication. To authorize an authentication, follow the procedure described in Configuring an SSL Profile for Use with RADIUS Server Hosts on page 75.

SCP

Using SCP to copy a digital certificate to a FastIron device

To copy trust certificate, client certificate, and client private key to an ICX device using secure copy, enter commands in the following format:

```
device# copy scp flash 1.1.1.1 /rootca.pem ssl-trust-cert
device# copy scp flash 1.1.1.1 /client.cert.pem ssl-client-cert
device# copy scp flash 1.1.1.1 /client-private.key.pem ssl-client-private-key
```

The example imports the digital certificate files for the trust certficate, client certificate, and client private key from the SCP server with the IP address 1.1.1.1.

Syntax: copy scp flash { *server-ip-address source-filename* **ssl-trust-cert** }

Syntax: copy scp flash { *server-ip-address source-filename* **ssl-client-cert** }

Syntax: copy scp flash { *server-ip-address source-filename* **ssl-client-private-key** }

SSHv2

Secure Shell version 2 (SSHv2) is allowed in FIPS mode.

The following SSH commands are affected when the FastIron device is in FIPS mode:

- The **ip ssh encryption aes-only** command is disabled.
- The **ip ssh key-authentication no** command is disabled.

- The **ip ssh scp** command ensures that SCP is enabled to run in FIPS mode. SCP is needed for file communication, and the **ip ssh scp disable** command is disabled in FIPS mode and displays the following message:

  ```
  FIPS Compliance: SCP needs to be enabled
  ```

- The **crypto key zeroize** command removes configured SSH keys.

  NOTE
  The following encryption methods are supported in FIPS mode:

  - aes256-ctr
  - aes128-ctr
  - aes256-cbc
  - aes128-cbc

  NOTE
  The following public key algorithms are supported in FIPS mode:

  1. ssh-rsa
  2. rsa-sha2-256
  3. rsa-sha2-512
  4. ecdsa-sha2-nistp384

  NOTE
  The following MAC algorithms are supported in FIPS mode:

  1. hmac-sha1
  2. hmac-sha2-256

  NOTE
  The following key exchange methods are supported in FIPS mode:

  1. ecdh-sha2-nistp256
  2. diffie-hellman-group14-sha256
  3. diffie-hellman-group16-sha512
  4. diffie-hellmangroup18-sha512
  5. ecdh-sha2-nistp384

Use the **show ip ssh config** command to display SSH configuration information.

SSH key generation time is affected by the increased security of authentication and encryption algorithms both in and out of FIPS mode.

Telnet

Telnet is disabled in FIPS mode as part of the default FIPS policy on the device. Attempts to start the Telnet server fail in FIPS mode.

TFTP

All **copy tftp** commands are disabled and return an error when TFTP operation is not allowed on the device in FIPS mode.

System Reset and Boot up in FIPS Mode

POST testing takes place as the device progresses through the boot sequence.

The following actions and limitations take effect when the device is operationally in FIPS mode according to the FIPS default policy:

- Boot up from TFTP is disabled.
- The monitor mode memory access command set is disabled. Configure an alternative FIPS policy to the default policy to access the command set. Refer to Modifying the FIPS Policy on page 36.
- Boot monitor access during a cold boot is disabled with the exception of the option to access monitor mode during the boot sequence.
- Access to memory test mode is disabled.
- Debug commands are disabled from the application prompt in FIPS mode.

Debugging in FIPS Mode

The device reloads automatically when it encounters a system reset and enters FIPS failure state. The cause of failure logs on the console and the device performs a self-reboot.

You can conduct debugging when a flexible FIPS policy is applied on the device. In the event of continuous reload, contact technical support.

Placing the Device in FIPS Mode

Placing the device in FIPS mode is a multiple-step process that begins with enabling FIPS mode on the device.

This places the device administratively in FIPS mode. To operate the device in FIPS mode, save the configuration, and reboot the device. Always back up the desired configuration to ensure it is saved in the event of a system reset.

General Steps to Place the Device in FIPS Mode

Perform the following steps to place the device in FIPS mode.

1. Copy the needed signature files. Refer to step 3 in Performing a FIPS or CC Software Upgrade to FastIron 09.0.10c on page 16.
2. Perform a FIPS self test to verify the correct signature files were copied. Refer to Performing a FIPS Self-test on page 35.
3. Assume Crypto Officer role.
4. Enable FIPS mode. Refer to Enabling FIPS Mode on page 29.
5. Zeroize shared secrets and host keys. Refer to Zeroizing Shared Secrets and Host Keys on page 32.
6. Save the configuration. Refer to Saving the Configuration on page 34.
7. Reload the device. Refer to Reloading the Device on page 34.

Enabling FIPS Mode

Perform the following steps to enable FIPS mode.

1. Attach a management station (PC or terminal) to the management module serial (console) port using a serial cable.

   When the device is not in a console session, FIPS-related commands return errors.

2. Verify that the device is in non-FIPS mode by using the **fips show** command.

```
device(config)# fips show
```

Syntax: fips show

The **fips show** command lists the current configuration of the device and can be run in both FIPS mode and non-FIPS modes to establish whether the device is truly in FIPS mode.

The output of the **fips show** command confirms that the device is in FIPS mode and identifies the device as either administratively or operationally in FIPS mode.

> **NOTE**
> If the FastIron device is in JITC mode, then you cannot enable FIPS on the device.

The following example shows the output of the **fips show** command before the **fips enable** command is entered. Administrative status and operational status are off.

```
device(config)# fips show
FIPS mode: Administrative Status: OFF, Operational Status: OFF
```

If the device is already in administrative FIPS mode, you can modify the FIPS policy. Refer to Modifying the FIPS Policy on page 36.

3. Use the **fips enable** command to place the device administratively in FIPS mode.

```
device(config)# fips enable
```

Syntax: [no] fips enable

The following example shows sample output of the **fips enable** command.

> **NOTE**
> Beginning with FastIron 08.0.20a, the RSA key pair is deleted when the FIPS mode is enabled. Use the **crypto key generate** command to generate the RSA key once the device is in FIPS mode.

```
device(config)# fips enable

FIPS: Deleting SSH Keys..RSA Key pair not found

FIPS: Disabling HTTP..HTTP already disabled

FIPS: Disabling Telnet..

FIPS: Disabling NTP..
FIPS: Disabling Tftp..
tftp disable set already.
This device is now running in FIPS administrative mode.
At this time you can alter this system's FIPS default security policy
and then enter FIPS operational mode.

Note: Making changes to the default FIPS security policy weakens
the security of the device and makes the device non-compliant with
FIPS 140-3 Level 1, design assurance Level 1
The default security policy defined in the FIPS
Security Policy Document ensures that the device complies with all
FIPS 140-3 specifications. Commands to alter the default security policy
are available to the crypto-officer; however, Ruckus Wireless does not recommend
making changes to the default security policy at any time.
====================================

To enter FIPS mode, complete the following steps:
1. Install the signature file now if not already done. Failure to install
signature or wrong signature file can cause continuous resets.
Also, optionally, configure FIPS policy commands that meets your network
requirements. You must explicitly configure the following services if you want
to use them when the device is operational in FIPS mode:
2. Enter the "fips zeroize all" command, which zeroes out the shared secrets
used by various networking protocols, including the host access passwords,
SSH and HTTPS host-keys with the digital signature based on the configured
FIPS Security Policy. If SSH ReKey Exchange value was not configured then
the default value of 30Mins and 500MB will be configured
3. Save the running configuration.
4. Reload the device.
5. Do not press "b" during reload, else FIPS or CC will not be enabled properly.
6. Enter the "fips show" command to verify that the device entered
FIPS or CC operational mode.
====================================

The system will disable the following services or commands after reload:
1. Telnet server will be disabled. The "telnet server" command will be removed.
2. SCP will be enabled. The "ip ssh scp disable" command will be removed.
3. HTTP server will be disabled. The "web-management http" command will be removed.
4. SNMP server will change as follows:
-SNMP support for v1 and v2 versions will be disabled.
-For SNMPv3 version md5 key and DES privacy password will be disabled.
5. NTP will be disabled.
Passwords/Keys which dont comply FIPS standards will be removed on reload.
aaa authentication method must be configured.
Disabling user when invalid password is entered is default in FIPS and above modes.
Default value of login recovery time is 3 secs.
No command sets the login recovery time to 3 secs and disables the user after 3 invalid attempts.
```

```
Default values of 3 attempts and 3 secs are not displayed in running config.
Please see FIPS config guide for complete details.
```

4. You can verify the status of the device as administratively in FIPS mode by using the **fips show** command.

   The following example shows the output of the **fips show** command on a FastIron device after the **fips enable** command is entered and administrative status is on and operational status is off:

   The following example shows the output of the **fips show** command on a CER devices after the fips enable command is entered and administrative status is on and operational status is off:

```
device# fips show
Cryptographic Module Version: FI-IP-CRYPTO
FIPS mode: Administrative status ON: Operational status OFF
Common-Criteria: Administrative status OFF: Operational status OFF
-----------------------------------------------------------------
 Some shared secrets inherited from non-fips mode may
 not be fips compliant and must be zeroized
 The system needs to be reloaded to operationally enter FIPS mode.
-----------------------------------------------------------------

System Specific:
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server: Disabled
Telnet client: Disabled
TFTP client: Disabled
SNMP Access to security objects: Disabled

Critical security Parameter updates across FIPS boundary:
Protocol Shared secret and host passwords: Clear
Password Display: Disabled


Certificate Specific:
HTTPS RSA Host Keys and Signature: Clear
SSH DSA Host keys: Clear
SSH RSA Host keys: Clear
CC Enable AAA Server Any: Clear
```

Zeroizing Shared Secrets and Host Keys

After you have reviewed the FIPS policy, use the **fips zeroize** command to zeroize the shared secrets and host keys used by various networking protocols.

```
device# fips zeroize all
```

Syntax: fips zeroize { **all** | **shared-secret** | **host-keys** | **pki-certs** }

The **all** option zeroizes all shared secrets and host keys. The **shared-secret** option zeroizes shared secret keys only. The **host-keys** option zeroizes host keys only.

For example, entering **fips zeroize shared-secret** zeroizes only the shared secret keys of various networking protocols and host access passwords.

> **NOTE**
> The **fips zeroize** command may cause operational failure within networking protocols using shared secrets and should be used with careful consideration.

The default FIPS policy calls for the zeroization of all keys using the **fips zeroize all** command option. When you apply a less strict FIPS policy than the default, zeroize at your discretion.

NOTE
The **fips zeroize all** command zeroizes all keys irrespective of the configured FIPS policy.

The following table lists the various keys used in the system that are zeroized in compliance with FIPS.

TABLE 3 Key Zeroization

Keys Used	Command Option Handling
DH private keys	Host-keys
FCSP Challenge Handshake Authentication Protocol (CHAP) secret	Host-keys
SSH session key	Host-keys
SSH RSA private key	Host-keys
RNG seed key	N/A
Passwords	Shared-secret
TLS private key	Host-keys
TLS pre-master secret	Host-keys
TLS session key	Host-keys
TLS authentication key	Host-keys
RADIUS secret	Shared-secret
Authentication passwords for various networking protocols	Shared-secret

Configuring User Authentication

RUCKUS FastIron devices support role-based authentication. A device can perform authentication and authorization (role selection) using RADIUS and local configuration database. FastIron devices also support multiple authentication methods for each service.

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (SSHv2, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local user authentication
- RADIUS authentication

When a list is configured, the device attempts the first method listed to provide authentication. If that method is not available (for example, the device cannot reach a RADIUS server), the device tries the next method until a method in the list is available or all methods have been tried.

FastIron devices allow multiple concurrent operators through SSHv2 and the console. One operator's configuration changes can overwrite the changes of another operator.

Local User Authentication

The local method of authentication uses a password associated with a user name to authenticate an operator. An operator enters a user name and corresponding password. The FastIron device assigns the role associated with the user name to the operator when authentication is successful.

To use local authentication, a Crypto-officer must define user accounts. The definition includes a user name, password, and privilege level (which determines the role).

RADIUS Authentication

The RADIUS method uses one or more RADIUS servers to verify user names and passwords. The FastIron device prompts an operator for user name and password. The device sends the user name and password to the RADIUS server. Upon successful authentication, the RADIUS server returns the operator's privilege level, which determines the operator's role. If a RADIUS server does not respond, the FastIron device sends the user name and password information to the next configured RADIUS server.

FastIron series devices support additional command authorization with RADIUS authentication. The following events occur when RADIUS command authorization takes place.

1. A user previously authenticated by a RADIUS server enters a command on the FastIron device.
2. The FastIron device looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the FastIron device looks at the list of commands returned to it when RADIUS server authenticated the user.

After RADIUS authentication takes place, the command list resides on the FastIron device. The device does not consult the RADIUS server again once the operator has been authenticated. This means that any changes made to the operator's command list on the RADIUS server are not reflected until the next time the RADIUS server authenticates the operator, and the server sends a new command list to the FastIron device.

NOTE
Radius over TLS is supported in FIPS mode.

To use RADIUS authentication, a Crypto-officer must configure RADIUS server settings along with authentication and authorization settings.

Saving the Configuration

NOTE
Keep a backup copy of the startup configuration in the event of system reset.

After zeroizing, use the **write memory** command to save the configuration.

```
device(config)# write memory
```

Reloading the Device

NOTE
Before upgrading to a new image in FIPS mode, ensure that the corresponding signature file is available in the flash memory.

After you have saved the configuration, reload the device using the **reload** command:

```
device# reload
```

Various tests, including Power-On Self Tests (POSTs) and Known Answer Tests (KATs), are run by the FastIron device during reload, during the transition between non-FIPS mode and FIPS mode.

POSTs check for the consistency of the FIPS-approved algorithms implemented on the device.

KATs are used to exercise various features of FIPS-approved algorithms.

All interfaces on the device are down until the tests are completed successfully.

Possible POST failure messages indicating that the device did not pass the tests successfully include the following messages:

```
Crypto module initialization and KNown Answer Test (KAT) failed with reason:(Error
Code 0x80000000)'CKR_VENDOR_DEFINED'
FIPS: Verifying Uboot Image Checksum ...Failed
```

```
FIPS: Verifying Image Checksum ... Failed
Checksum Verification Failed For Package: AAAProcessPkg
```

If there is a failure while the POSTs are being run, the device reboots. Monitor mode can be accessed to troubleshoot the issue.

After all tests are completed successfully, the device reloads in FIPS mode and FIPS mode is successfully enabled and operational on the FastIron device.

You can verify the status of the device as operationally in FIPS mode by using the **fips show** command.

```
device(config)# fips show
```

The following example shows **fips show** command after the device reloads successfully in the default strict FIPS mode. Administrative status and operational status are on.

```
device# fips show
Cryptographic Module Version: FI-IP-CRYPTO
FIPS mode: Administrative status ON: Operational status ON
Common-Criteria: Administrative status OFF: Operational status OFF

System Specific:
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server: Disabled
Telnet client: Disabled
TFTP client: Disabled
SNMP Access to security objects: Disabled

Critical security Parameter updates across FIPS boundary:
Protocol Shared secret and host passwords: Clear
Password Display: Disabled

Certificate Specific:
HTTPS RSA Host Keys and Signature: Clear
SSH DSA Host keys: Clear
SSH RSA Host keys: Clear
CC Enable AAA Server Any: Clear
```

Performing a FIPS Self-test

Use the FIPS self-test to verify the sanity of FIPS software.

For more information on the FIPS self-test, refer to Running FIPS Self-tests on page 37.

> **NOTE**
> During FIPS self-test, the CPU usage is high. The **fips self-tests** command should only be used prior to the device being placed into FIPS operational or administrative modes. Using the **fips self-tests** command in FIPS operational or administrative mode may result in the device rebooting as per the FIPS criteria.

- From the Privileged EXEC level of the CLI on the console, use the **fips self-tests** command to verify that the FIPS Software and Firmware Integrity Test passes.

  Syntax: fips self-tests

- The following example shows the FIPS Software and Firmware Integrity Test as passed:

```
device# fips self-tests
Running FIPS Power On Self Tests and Software/Firmware Integrity Test.
FIPS Power On Self Tests and Software/Firmware Integrity tests successful.
.....<output truncated>.......
```

If the test fails, make sure that the correct signature file was copied for the correct image file and version, and recopy as needed.

NOTE
The FIPS self-test must pass before the configuration is saved and the device is reloaded.

Modifying the FIPS Policy

After the device is administratively in FIPS mode, you can modify the default FIPS policy.

NOTE
Making changes to the default FIPS policy on the device is not recommended and weakens the security of the device. Any modification of the default FIPS policy places the device in a state that is not in compliance with FIPS 140-3.

The output of the **fips enable** command displays which protocols that constitute the FIPS policy are set in compliance with FIPS standards by default and can be adjusted to set a more flexible policy. The remaining protocols that constitute the FIPS policy are set to the appropriate status automatically during reload due to the **fips enable** command. The default FIPS policy is detailed in How FIPS Works on page 13.

When you make no changes to the FIPS policy, the default FIPS policy is applied on the device and the device operates in strict FIPS mode upon reload, in full compliance with FIPS 140-3 specifications.

To set a more flexible FIPS policy on the FastIron device, use the following commands as desired to modify the default FIPS policy.

- Allow TFTP access:

  ```
  device(config)# fips policy allow tftp-access
  ```

  Syntax: [no] fips policy allow tftp-access

- Allow SNMP access to the critical security parameter (CSP) MIB objects:

  ```
  device(config)# fips policy allow snmp-csp-access
  ```

  Syntax: [no] fips policy allow snmp-csp-access

- Allow access to monitor mode for debugging both from application and boot prompts:

  ```
  device (config)# fips policy allow monitor-full-access
  ```

  Syntax: [no] fips policy allow monitor-full-access

  NOTE
  During an application reset, monitor access is restored to allow debugging.

- Retain the shared secret keys for all protocols and the host passwords:

  ```
  device(config)# fips policy retain shared-secrets
  ```

  Syntax: [no] fips policy retain shared-secrets

- Retain the SSH DSA host keys:

  ```
  device(config)# fips policy retain dsa-host-keys
  ```

  Syntax: [no] fips policy retain dsa-host-keys

- Retain the HTTPS RSA host keys and the HTTPS server digital certificate:

  ```
  device(config)# fips policy retain rsa-host-keys
  ```

  Syntax: [no] fips policy retain rsa-host-keys

Disabling FIPS Mode

NOTE
Even after disabling FIPS mode, you should always load the signature file before loading the software image file.

NOTE
If you disable FIPS mode, all local users are removed. Removal of the last local user triggers the removal of AAA configuration specific to local authentication.

Use the **no fips enable** command to disable FIPS mode on the FastIron device.

```
device(config)# no fips enable
```

After you enter the command, a warning displays that FIPS mode will be disabled.

This command performs the following policy-related operations:

- Enables TFTP access.
- Re-enables SNMP access to critical security parameter (CSP) MIB objects.
- Re-enables SNMPv3 encryption protocol DES for future SNMPv3 user configuration.
- Re-enables access to monitor mode.
- Zeroizes shared secrets, SSH and HTTPS host keys, and the HTTPS certificate based on the configured FIPS policy.

The **no fips enable** command also performs the non-policy-related operation of re-enabling the RC4 cipher for the HTTPS server.

Changes to the running configuration are not saved to the startup configuration; therefore, when the device reloads, it returns to FIPS mode.

Use the **write memory** command to save the running configuration.

Running FIPS Self-tests

Use the **fips self-tests** command either in FIPS mode or non-FIPS mode to run the Known Answer Tests (KATs) and conditional tests on demand in both FIPS mode and non-FIPS mode.

```
device# fips self-tests
```

Syntax: fips self-tests

The following log message is generated when the KAT is completed, but no trap messages are generated because the system is not fully operational.

```
“Crypto module initialization and Known Answer Test (KAT) passed”.
```

Common Criteria Certification

Common Criteria Overview

Common Criteria certification for a device enforces a set of security standards and feature limitations on a device to be compliant with the Common Criteria standards, similar to placing the device in FIPS mode. These restrictions are in addition to the requirements of FIPS mode. When the device is placed in Common Criteria mode, several security features that are available in FIPS mode are unavailable on the device. Because Common Criteria mode enforces security restrictions additional to FIPS mode, procedures and information are provided in relation to those for the FIPS mode.

For information about enabling FIPS mode on the device, refer to FIPS Configuration on page 21.

For additional information on features available in both FIPS and CC mode and their configuration, refer to the related FIPS sections.

For information on SSH, refer to the following sections:

- SSH on page 22
- SSH Clients on page 24
- Usernames and SSH Public Key Authentication on page 24.

For information on self-tests, refer to Running FIPS Self-tests on page 37.

NOTE
Common Criteria mode becomes available once a device is FIPS-enabled.

NOTE
To determine if the FastIron device and current software version is Common Criteria-certified, refer to https://www.niap-ccevs.org/Product/index.cfm. The Security Targets identified in the RUCKUS PCL entries define the scope of features that were evaluated. Refer to the release notes for the software version running on the device to verify that the software is FIPS- and Common Criteria-certified.

NOTE
MACsec is supported on ICX 7450, ICX 7550, ICX 7650, and ICX 7850 devices.

The following table summarizes support for Common Criteria protection profiles by FastIron device.

TABLE 4 Common Criteria protection profiles supported by device

Platform	NDcPP2.2e	MACsec (ndcpp_macsec_ep_v1.2)
ICX7150	Yes	No
ICX7250	Yes	No

TABLE 4 Common Criteria protection profiles supported by device (continued)

Platform	NDcPP2.2e	MACsec (ndcpp_macsec_ep_v1.2)
ICX7450	Yes	Yes
ICX7550	Yes	Yes
ICX7650	Yes	Yes
ICX7850	Yes	Yes

You can enable Common Criteria mode on a device directly from non-FIPS mode, or on a device already in FIPS mode. The following table summarizes the transitions.

TABLE 5 Transition to Common Criteria mode

From	To non-FIPS mode	To FIPS mode	To Common Criteria mode
Non-FIPS mode	Not applicable	Use the **fips enable** command	Use the **fips enable common-criteria** command
FIPS mode	Use the **no fips enable** command	Not applicable	Use the **fips enable common-criteria** command
Common Criteria mode	Use the **no fips enable** or **no fips enable common-criteria** command	Use the following commands in a sequence: 1. **no fips enable** 2. **reload device** 3. **fips enable**	Not applicable

Be advised of the following considerations:

- Disabling FIPS mode from the Common Criteria mode using the **no fips enable** command downgrades the device directly into non-FIPS mode.
- You cannot directly transition from Common Criteria mode to FIPS mode. To transition to FIPS mode, you must disable FIPS mode, reload the device, and then enable FIPS mode.

Features Unavailable in FIPS and Common Criteria Mode

Some of the security features are disabled in FIPS/CC mode:

- SSHv2: Host and client key generation methods using DSA and the RSA-1024 key size are not supported (only RSA 2048 and higher key sizes are supported). Therefore, the following commands are not supported:
  - **crypto key gen rsa modulus 1024**
  - **crypto key zero rsa modulus 1024**
- TLS and HTTPS: The RSA 1024 key size for SSL or TLS private key generation is not supported (FastIron devices support only 2048 and above key sizes).
- SSH key exchange: The SSH key exchange method diffie-hellman-group1-sha1 and diffie-hellman-group14-sha1 are not supported. Only the following SSH key exchange methods are supported:
  - diffie-hellman-group14-sha256
  - diffie-hellman-group16-sha512
  - diffie-hellman-group18-sha512
  - ecdh-sha2-nistp256
  - ecdh-sha2-nistp384

Features Available in Common Criteria Mode

The following features are available in Common Criteria mode.

1. Secure Syslog: The secure syslog feature uses TLS to securely send the log messages to the log server.
2. Radius Authentication over a secure tunnel: All user authentication uses a TLS tunnel to communicate with the Radius server to authenticate the user.

Supported Algorithms for SSH Client

An SSH client can connect to a device in Common Criteria mode only with the AES-128 and AES-256 encryption algorithms.

Supported Cipher Suites

The following mandatory cipher suites are supported for TLS 1.1 and TLS 1.2 in Common Criteria mode.

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

RADIUS Protocol in CC Mode

HMAC-MD5 authentication used in RADIUS is allowed in CC mode.

RADIUS allows client-to-server authentication. To authorize an authentication, follow the procedure described in Configuring an SSL Profile for Use with RADIUS Server Hosts on page 75.

SCP for Common Criteria

Use SCP as described in the FIPS section SCP on page 27.

Enabling Common Criteria Mode

When you enable Common Criteria mode on the device, it enters the Common Criteria Administrative mode. Similar to FIPS, Common Criteria also has administrative and operational modes:

- Common Criteria Administrative mode: Log in to the device console and enable the Common Criteria mode. You can optionally modify the default Common Criteria security policy in this mode.

  NOTE
  When you execute the command to reload the device, the validation of the software image is triggered. If verification fails, the device continuously reboots after device reload.

- Common Criteria Operational mode: Transition to Common Criteria operational mode from Common Criteria Administrative mode. After you transition the device to the Operational mode, you must save the configuration and reboot the device.

Entering Common Criteria Administrative Mode

Use the following command to enter Common Criteria mode.

```
device(config)# fips enable common-criteria
```

Syntax: [no] fips enable common-criteria

The following example includes the detailed banner that is displayed after you enter the command.

```
device# configure
device(config)# fips enable common-criteria
CLI-FIPS: Common Criteria is enabled with FIPS status Disabled

FIPS: Disabling HTTP..HTTP already disabled

FIPS: Disabling Telnet..
Error - No active TELNET sessions

FIPS: Disabling Tftp..

FIPS: Deleting SSH Keys..RSA Key pair is successfully deleted

FIPS: Disabling HTTP..HTTP already disabled

FIPS: Disabling Telnet..

SYSLOG: <14> Apr 15 12:50:53 ICX7450-48P Router Crypto: RSA Key pair is successfully deleted
Error - No active TELNET sessions

FIPS: Disabling Tftp..
tftp disable set already.

FIPS: Deleting SSH Keys..RSA Key pair not found
This device is now running in CC administrative mode.
At this time you can alter this system's CC default security policy
and then enter CC operational mode.

Note: Making changes to the default policy makes the device non-compliant
with CC and FIPS 140-3 Level 1, design assurance Level 1
The default security policy defined in the FIPS
Security Policy Document ensures that the device complies with all
FIPS 140-3 specifications. Commands to alter the default security policy
are available to the crypto-officer; however, Ruckus Wireless does not recommend
making changes to the default security policy at any time.
====================================

To enter CC mode, complete the following steps:
1. Optionally, configure FIPS policy commands that meets your network
   requirements. You must explicitly configure the following services if you
   want to use them when the device is operational in CC mode:
2. Enter the "fips zeroize all" command, which zeroes out the shared secrets
    used by various networking protocols, including the host access passwords,
    SSH and HTTPS host-keys with the digital signature based on the configured
    FIPS Security Policy. If SSH ReKey Exchange value was not configured then
    the default value of 30Mins and 500MB will be configured
3. Save the running configuration.
4. Reload the device.
5. Do not press "b" during reload, else FIPS or CC will not be enabled properly.
6. Enter the "fips show" command to verify that the device entered
    FIPS or CC operational mode.
====================================

The system will disable the following services or commands after reload:
1. Telnet server will be disabled. The "telnet server" command will be removed.
2. SCP will be enabled. The "ip ssh scp disable" command will be removed.
3. HTTP server will be disabled. The "web-management http" command will be removed.
Passwords/Keys which dont comply FIPS standards will be removed on reload.
aaa authentication method must be configured.
Disabling user when invalid password is entered is default in FIPS and above modes.
```

```
Default value of login recovery time is 3 secs.
No command sets the login recovery time to 3 secs and disables the user after 3 invalid attempts.
Default values of 3 attempts and 3 secs are not displayed in running config.
Please see FIPS config guide for complete details.
====================================
Additionally, in CC mode, the system will disable the following
services or commands after reload:
UDP Syslog servers will be deleted from configuration(only in the CC operational mode).
DSA keys will be deleted from configuration, and will be disabled .
RSA key sizes will be restricted to 2048 and above in the configuration.
Non-TLS TACACS+ servers will be disabled from configuration.
For SSH Key Exchange, only diffie-hellman-group14 algorithm is allowed.
```

SSH Rekey Exchange

In SSH2 implementation, if an SSH session is authenticated and established, the session remains connected until the user closes it or until it is closed after the configured idle time limit. Prolonged usage of the session key negotiated at connection startup poses several security issues and exposes SSH connections to man-in-middle attacks. To safeguard existing SSH connections from security vulnerabilities, new keys should be exchanged frequently.

SSH rekeying is the process of exchanging the session keys at a configured interval, based on a time limit or a data limit for the SSH session. SSH rekeying is triggered when the time limit (maximum minutes) or the data limit has been reached for the session. Rekey can be initiated by either the client or the server. While the key exchange renegotiation is taking place, data does not pass through the SSH connection. The algorithm that was used at connection startup is used during rekey.

In FIPS and CC modes, the SSH rekey feature is enabled by default and cannot be disabled. The default value for time is 30 minutes, and the default limit for data is 500 Mbytes in both FIPS and CC modes. If the rekey configuration is removed in either FIPS or CC mode, the default values are applied. The default values are not displayed in the configuration.

When moving from non-FIPS mode to FIPS/CC mode:, if SSH rekey is enabled in non-FIPS mode, the configured values are applied while moving to FIPS mode. If SSH rekey is not configured in non-FIPS mode, the default values in FIPS and CC mode will be applied.

In transitioning from FIPS or CC mode to non-FIPS mode, the SSH rekey configuration is removed, and the feature is disabled in non-FIPS mode.

SSH Rekey Configuration Notes

- The encryption method must not be modified during the rekey process.
- When rekey configuration has changed, the change has no impact on the existing session until the next rekey exchange for the session occurs.
- When a rekey exchange occurs, the value of data and time for the corresponding SSH session is reset to the configured rekey value.
- SSH sessions established without rekey configuration do not have the rekey functionality.
- When rekey is enabled, the existing SSH session does not have the rekey functionality until the rekey exchange occurs from the other side.
- When the rekey configuration is removed, the default values are applied.

SSH Rekey Configuration Examples

The following example configures rekeying of the outbound SSH session every hour.

```
device# configure terminal
device(config)# ip ssh rekey client time 60
```

The following example configures rekeying on the inbound SSH session whenever 10,000 Kilobytes of data are transmitted.

```
device# configure terminal
device(config)# ip ssh rekey server data 10000
```

The following example resets SSH rekey exchange to default settings (and does not disable the function). The defaults can be restored from either the client or the server side.

```
device# configure terminal
device(config)# no ip ssh rekey client time 60
```

Syntax: ip ssh rekey { **client** | **server** } { **data** *Kbytes* | **time** *minutes* }

Syntax: no ip ssh rekey { **client** | **server** } { **data** *Kbytes* | **time** *minutes* }

CLI Banner Configuration

FastIron devices can be configured to display a greeting message on users' terminals when they enter the Privileged EXEC CLI level.

Setting a Message of the Day Banner

Use the **banner motd** command to configure the FastIron device to display a message on a user terminal when a CLI session (for example, SSH, console) is established. The banner motd command allows you to define a delimiting character to be used at the beginning and end of the text to be used as the banner. The delimiting character cannot appear in the message and can be any character except a double quotation mark ("). The dollar sign ($) is used as the delimiting character in the example. The banner text can be up to 4,000 characters long and can contain multiple lines.

For example, to display the message "Welcome to ICX!" when a Telnet CLI session is established, enter the following commands.

```
Device# configure terminal
Device(config)# banner motd $
Enter text message. End with character '$'.
Welcome to ICX!$
Device(config)#
```

Use the **no banner motd** command to remove the banner.

Entering Common Criteria Operational Mode

When the device is in Common Criteria Administrative mode, perform the following steps to place the device into Common Criteria Operational mode.

1. Configure the local user accounts as secure and delete non-secure user accounts. A local user account is secure when it has a password with characters from three or more character classes. These character classes are uppercase, lowercase, numeric, and ASCII non-alphanumeric characters.
2. Configure secure logging by setting up the encrypted syslog server. For details, refer to Encrypted Syslog Servers in Common Criteria Mode on page 45.
3. Enable AAA authentication with the **aaa authentication login default** command.
4. Use **logging cli-command** to allow you to log all syntactically valid CLI commands from each user session into the system log.
5. Configure a motd banner using the **banner motd** command. Refer to CLI Banner Configuration on page 44 for more information.
6. Use the **write memory** command to save the configuration.
7. Use the **reload** command to reload the device.

On successful completion of these steps, the device will be in Common Criteria Operational mode.

> **NOTE**
> FIPS self-tests are executed as part of device bootup. If any of the self-tests fails, the device reloads automatically. If there are continuous reloads, please contact Ruckus technical support.

NOTE
Use the **exit** command to terminate a local or remote interactive session.

Displaying Common Criteria Information

After you have enabled Common Criteria Administrative mode on the device, you can display the information with the **fips show** command.

After you have enabled Common Criteria operational mode by reloading the device, enter the **fips show** command to verify the operational mode status:

```
device# fips show
FIPS mode: Administrative status ON: Operational status ON
Common-Criteria: Administrative status ON: Operational status ON
System Specific
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server: Disabled
Telnet client: Disabled
TFTP client: Disabled
SNMP Access to security objects: Disabled

Critical security Parameter updates across FIPS boundary:
Protocol Shared secret and host passwords: Clear
Password Display: Disabled

HTTPS RSA Host Keys and Signature: Clear
SSH DSA Host keys: Clear
SSH RSA Host keys: Clear
```

Encrypted Syslog Servers in Common Criteria Mode

FastIron devices in any mode send the generated syslog messages in real time to the local log storage on the device. Local log storage can be configured and holds up to 4,000 messages by default. Old audits are overwritten in local log storage when the configured maximum is reached. FastIron devices in any mode also send generated syslog messages to a syslog server (only if a syslog server is configured and available).

A FastIron device running in Common Criteria operational mode queues the syslog messages if a syslog server is not available or configured for the device. This queue is not related to the local syslog messages store and it is cleared when the syslog messages in the queue are forwarded to the syslog server. The queue cannot hold more than 3,000 syslog messages. On reaching the maximum message limit, the device displays an error message and no further syslog messages are queued.

Parameters that are defined for syslog server connections, such as specifying the hold time for queued messages and traps when the device reloads or switches over, are applicable for encrypted syslog connections as well.

The following table summarizes the transitions to and from Common Criteria mode.

TABLE 6 Syslog Server Connections during Transition to and from Common Criteria Mode

From	To Non-FIPS Mode	To FIPS Mode	To Common Criteria Operational Mode
Non-FIPS mode	Not applicable	No change. FIPS mode does not support encrypted syslog servers.	Both UDP-based and encrypted syslog server connections are allowed in CC Operational mode.
FIPS mode	No change	Not applicable	Both UDP-based and encrypted syslog server connections are allowed in CC Operational mode.

TABLE 6 Syslog Server Connections during Transition to and from Common Criteria Mode (continued)

From	To Non-FIPS Mode	To FIPS Mode	To Common Criteria Operational Mode
Common Criteria mode	All the SSL servers are removed. Non-FIPS mode does not support encrypted syslog server connections.	Not allowed. You must disable Common Criteria mode to revert to non-FIPS mode, and then re-enable FIPS mode. FIPS mode does not support encrypted syslog server connections.	Not applicable

Configuring the Logging Buffer for Local Storage in Common Criteria Mode

Use the logging buffered command to configure the size of the local syslog message buffer. By default, the buffer holds 4,000 messages. You can configure the buffer to hold from 1 through 4,000 messages. You can also configure the system to send a notification when the buffered messages reach a specified percentage of the maximum.

The **no** form of the logging buffered command returns the buffer size to its default value.

NOTE
Informational and debugging messages are not logged.

NOTE
When you change the logging buffer size, you must save the configuration and reload the system for the change to take effect. Any configuration change clears the logging buffer.

Examples:

The following example configures the local syslog buffer to retain 1,500 messages before overwriting.

```
ICX# configure terminal
ICX(config)# logging buffered 1500
```

Syntax: logging buffered *total_messages*

The following example sets a percentage full threshold for buffered messages at 90%. When the threshold is exceeded, a system warning message is generated.

```
ICX(config)# logging buffered threshold 90
```

Syntax: logging buffered threshold *percentage*

AAA Servers in Common Criteria Mode

Common Criteria mode requires that devices support NDcPP version 2.2e or above. This standard requires the communication of the device with AAA servers to take place over a TLS-encrypted session.

Even though you can configure multiple TLS-encrypted RADIUS servers, only one connection can be active at any time. If another TLS-encrypted RADIUS session is attempted at the same time as the first RADIUS session, the connection attempt is rejected.

When the device is in Common Criteria Operational mode, and the device has been configured for a TLS encrypted RADIUS server for authentication, only one administrator is able to administer the device. In addition, accounting and authorizing using the TLS-encrypted RADIUS server are disabled.

Modifying the Common Criteria Policies to Use Non-encrypted AAA Servers

If required, you can modify the Common Criteria policies to allow AAA servers that do not use TLS encryption to be configured, such as RADIUS servers. When non-encrypted AAA servers are allowed, you cannot configure TLS-encrypted TACACS+ servers on the device.

> **NOTE**
> Modifying the default Common Criteria policy makes the device noncompliant with Common Criteria standards.

To allow any AAA server to work with the device in Common Criteria mode, enter the following command:

```
device# fips policy allow common-criteria aaa-server-any
```

Syntax: [no] fips policy allow common-criteria aaa-server-any

Use the **[no]** form of the command to remove non-encrypted AAA servers. If any non-encrypted AAA servers were available on the device, they are removed when Common Criteria mode is enabled on the device.

Downgrading from Common Criteria Mode to Non-FIPS Mode

Downgrading a device from Common Criteria mode either to FIPS mode or to non-FIPS mode uses the same command. You cannot directly downgrade to FIPS mode. You must first downgrade to non-FIPS mode and then enable FIPS mode using the procedures detailed in the previous chapter.

After the device is placed in non-FIPS mode, you can use SCP to download and initialize an older image. Use the following steps to revert to a non-FIPS-compliant image.

1. Log in to the device by entering your username and password.
2. Disable Common Criteria mode by entering the **no fips enable** or **no fips enable common-criteria** command.
3. Regenerate SSH host keys or other shared secrets as needed for access after reload.
4. To replace the startup configuration with the **no fips enable** configuration, enter the **write memory** command.
5. Reload the configuration by entering the **reload** command.

Commercial Solutions for Classified program

The Commercial Solutions for Classified (CSfC) program was established by the United States government to enable commercial networking applications to be used in layered solutions that protect classified National Security Systems (NSS) data. The CSfC program provides the ability to communicate securely based on commercial standards in a solution. RUCKUS, as a networking company, supports CSfC and many of the products listed in the CSfC component list. RUCKUS devices must be approved by the CSfC program to be deployed in government or federal networks.

To determine if the FastIron device and current software version are CSfC certified, refer to the following URL:

https://www.nsa.gov/Resources/Commercial-Solutions-for-Classified-Program/.

Configuring NTP

NTP services are disabled on all interfaces by default.

Before you begin to configure NTP, you must use the **clock set** command to set the time on your device to within 1,000 seconds of the Coordinated Universal Time (UTC).

> **NOTE**
> Multicast and broadcast NTP packets are ignored by default.

Enabling NTP

To enable NTP, use the **ntp** command in global configuration mode. This command enables the NTP client mode and server mode.

```
device(config)# ntp
device(config-ntp)#
```

Use the **no** form of the command to disable NTP and remove the NTP configuration.

> **NOTE**
> The **no ntp** command removes all the NTP configuration configured statistically as well as learned associations from NTP neighbors.

Disabling NTP

To disable the NTP server and client modes, use the **disable** command in NTP configuration mode. Disabling the NTP server or client mode does not remove the configuration.

```
device# configure terminal
device(config)# ntp
device(config-ntp)# disable
```

To enable the client mode, use the **no disable** command. To enable the client and server mode, use the **no disable serve** command.

The **serve** keyword disables NTP server mode. If the **serve** keyword is specified, NTP does not serve the time to downstream devices. In contrast, if the **serve** keyword is not specified, both NTP client and NTP server modes are disabled.

> **NOTE**
> The **no disable** command enables both the client and the server if the client was already enabled and the server was already disabled at that time the **no disable server** command was entered.

Enabling NTP Authentication

To enable Network Time Protocol (NTP) strict authentication, use the **authenticate** command. To disable authentication, use the **no** form of the command.

By default, authentication is disabled.

```
device(config-ntp)# authenticate
```

Defining an Authentication Key

To define an authentication key for Network Time Protocol (NTP), use the **authentication-key** command. To remove the authentication key for NTP, use the **no** form of the command.

By default, authentication keys are not configured.

```
device(config-ntp)# authentication-key key-id 1 sha1 A3RuZcbxpG
```

The key string is the value of the SHA1 key. The maximum length of the key string is 16 characters. Up to 32 authentication keys can be defined.

Configuring the NTP Client

To configure the device in client mode and specify the NTP servers to synchronize the system clock, use the **server** command. A maximum of eight NTP servers can be configured. To remove the NTP server configuration, use the **no** form of the command.

By default, no servers are configured.

Examples:

The following example configures the NTP server in NTP version 3 mode.

```
device(config-ntp)# server 1.2.3.4 version 3 key 1234 minpoll 4

device(config-ntp)# server 1:2::3:4 version 3 key 1234 minpoll 4
```

Syntax: server { *ipv4_address* | *ipv6_address* **version 3 key** *key_id* **minpoll** *interval* }

The minpoll interval specifies the shortest polling interval. It is expressed as a power of 2 and can be in the range 4 through 17.

The following example configures the NTP server in NTP version 4 mode.

```
device(config-ntp)# server 1.2.3.4 key 1234

device(config-ntp)# server 1:2::3:4 key 1234
```

Syntax: server { *ipv4_address* | *ipv6_address* **key** *key_id* }

Displaying NTP Status

Use the **show ntp status** command to display the NTP status.

```
device# show ntp status
Clock is synchronized, stratum 4, reference clock is 10.20.99.174
precision is 2**-16
reference time is D281713A.80000000 (03:21:29.3653007907 GMT+00 Thu Dec 01 2011)
clock offset is -2.3307 msec, root delay is 24.6646 msec
root dispersion is 130.3376 msec, peer dispersion is 84.3335 msec
system poll interval is 64, last clock update was 26 sec ago
NTP server mode is enabled, NTP client mode is enabled
NTP master mode is disabled, NTP master stratum is 8
NTP is not in panic mode
```

Displaying NTP Association Information

Use the **show ntp associations** command to display detailed association information for the NTP server or peers.

```
device# show ntp associations
address ref clock st when poll reach delay offset disp
*~172.19.69.1 172.24.114.33 3 25 64 3 2.89 0.234 39377
~2001:235::234
INIT 16 - 64 0 0.00 0.000 15937
* synced, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Displaying NTP Association Details

Use the **show ntp associations detail** command to display association information for all NTP servers and peers.

```
device# show ntp association detail
2001:1:99:30::1 configured server, sys peer, stratum 3
ref ID 204.235.61.9, time d288dc3b.f2a17891 (10:23:55.4070668433 Pacific Tue Dec 06 2011)
our mode client, peer mode server, our poll intvl 10, peer poll intvl 10,
root delay 0.08551025 msec, root disp 0.09309387, reach 17, root dist 0.17668502
delay 0.69961487 msec, offset -13.49459670 msec, dispersion 17.31550718,
```

```
precision 2**-16, version 4
org time d288df70.a91de561 (10:37:36.2837308769 Pacific Tue Dec 06 2011)
rcv time d288df70.a0c8d19e (10:37:36.2697515422 Pacific Tue Dec 06 2011)
xmt time d288df70.a086e4de (10:37:36.2693194974 Pacific Tue Dec 06 2011)
filter delay 1.7736 0.9933 0.8873 0.6699 0.7709 0.7712 0.7734 6.7741
filter offset -17.9936 33.0014 -13.6604 -13.4494 -14.4481 -16.4453 -18.4423 -22.0025
filter disp 15.6660 0.0030 17.7730 17.7700 17.6670 17.6640 17.6610 16.6635
filter epoch 55824 56866 55686 55688 55690 55692 55694 55759
```

NTP Client Mode Configuration Example

The following example configures the RUCKUS device in NTP client mode.

```
device(config-ntp)# server 10.1.2.3 minpoll 5 maxpoll 10
device(config-ntp)# server 11::1/24
device(config-ntp)# peer 10.100.12.83
device(config-ntp)# disable serve
```

NTP Strict Authentication Configuration Example

The following example configures the RUCKUS device in strict authentication mode.

```
device(config-ntp)# authenticate
device(config-ntp)# authentication-key key-id 1 sha1 A3RuZcbxpG
device(config-ntp)# server 10.1.2.4 key 1
```

Configuring PKI

You can create PKI entities for use in certificate authentication. To participate in certificate authentication with a Certificate Authority (CA), PKI entities must be enrolled. Entities can be enrolled through an automatic enrollment process, which allows them to send a certificate signing request (CSR) and to receive the required X.509 certificates from the CA in response. Entities can also be enrolled manually as described in the section "PKI manual import." The following procedure explains how to configure automatic enrollment.

Configuring PKI involves the following tasks:

- Generate a cryptographic key using either the **ec** (elliptical key pair) or the **rsa** key pair option for PKI.
- Create a PKI entity.
- Configure the PKI profile.
- Configure the PKI trustpoint.
- Authenticate the PKI.
- Enroll the PKI.

Perform the following steps to complete these tasks.

1. Create a cryptographic key as shown in the following example.

   The first example generates a key pair using the rsa option for the PKI. The second example generates an elliptical key pair for the PKI.

   ```
   device# configure terminal
   device(config)# crypto key generate rsa label < name >

   device# configure terminal
   device(config)# crypto key generate ec label < name >
   ```

2. Enter PKI entity configure submode to configure end user parameters.

   NOTE
   PKI entity configuration is used for auto-enrollment only.

   The following example enters configuration submode for the PKI entity named entity1.

```
device (config)# pki entity entity1
device(config-pki-entity-entity1)#
```

3. Configure PKI entity details, including common name, country name, state name, and organization name. Country names use a two-letter abbreviation.

   NOTE
   It is recommended that you use quotes around text strings. Quotes are required when a name includes a space.

   The first example below provides command syntax for entering PKI entity details.

   The second example configures realistic parameters for entity1.

```
device(config-pki-entity-entity1)# common-name < name >
device(config-pki-entity-entity1)# country-name < country-name >
device(config-pki-entity-entity1)# state-name < state-name >
device(config-pki-entity-entity1)# org-unit-name < unit-name >
device(config-pki-entity-entity1)# org-name < org-name >
device(config-pki-entity-entity1)# email-id < email-address >
device(config-pki-entity-entity1)# location < location-name >
device(config-pki-entity-entity1)#
device(config-pki-entity-entity1)# exit

device(config-pki-entity-entity1)# common-name "tester1"
device(config-pki-entity-entity1)# country-name "IN"
device(config-pki-entity-entity1)# state-name "KA"
device(config-pki-entity-entity1)# org-unit-name "FI"
device(config-pki-entity-entity1)# org-name "Ruckus"
device(config-pki-entity-entity1)# email-id "user@ruckus.com"
device(config-pki-entity-entity1)# location "BG"
device(config-pki-entity-entity1)#
device(config-pki-entity-entity1)# exit
```

4. Configure the PKI enrollment profile for use later in the enrollment process, including the following items:
   - Profile name
   - An authentication URL for the CA server where authentication requests are sent (for automatic enrollment only)
   - An enrollment URL for the CA server where enrollment requests are sent (for automatic enrollment only)
   - The challenge password obtained from the CA

   The following example provides profile enrollment syntax.

   ```
   device(config)# pki profile-enrollment < profile-name >
   device(config-pki-profile-enrollment-profile1)# authentication-url < URL >
   device(config-pki-profile-enrollment-profile1)# authentication-command < command >
   device(config-pki-profile-enrollment-profile1)# enrollment-url < URL >
   device(config-pki-profile-enrollment-profile1)# password < password >
   ```

   The following example configures the PKI enrollment profile named profile1.

   ```
   device(config)# pki profile-enrollment profile1
   device(config-pki-profile-enrollment-profile1)# authentication-url http://WIN-
   N6C3R0LUDAJ.englab.ruckus.com/CertSrv/mscep/mscep.dll
   device(config-pki-profile-enrollment-profile1)# authentication-command WIN-
   N6C3R0LUDAJ.englab.ruckus.com_englab-WIN-N6C3R0LUDAJ-CA-15
   device(config-pki-profile-enrollment-profile1)# enrollment-url http://WIN-
   N6C3R0LUDAJ.englab.ruckus.com/CertSrv/mscep/mscep.dll
   device(config-pki-profile-enrollment-profile1)# password DB6E1F091AEF0244
   device(config-pki-profile-enrollment-profile1)# exit
   ```

5. Configure the trustpoint name and details, including the following items:
   - The enrollment option (automatic)
   - Enrollment retry-period (1 through 60 minutes)
   - Name of enrollment profile to used in the enrollment process
   - Name of the pre-configured PKI entity to be enrolled
   - Key pair type and label (for key generated previously using crypto commands)
   - Digital fingerprint for rootca (obtained from the rootca certificate)
   - OCSP transport protocol (HTTP) and method (post)

   The following example provides PKI trustpoint command syntax.

   ```
   device(config)# pki trustpoint < trustpoint-name >
   device(config-pki-trustpoint-trust1)# auto-enroll
   device(config-pki-trustpoint-trust1)# enrollment retry-period < number >
   device(config-pki-trustpoint-trust1)# enrollment profile < profile-name >
   device(config-pki-trustpoint-trust1)# pki-entity < entity-name >
   device(config-pki-trustpoint-trust1)# { eckeypair | rsakeypair } key-label < label >
   device(config-pki-trustpoint-trust1)# fingerprint < fingerprint-value >
   device(config-pki-trustpoint-trust1)# ocsp http post
   device(config-pki-trustpoint-trust1)# exit
   ```

   The following example configures the PKI trustpoint named trust1.

   ```
   device(config)# pki trustpoint trust1
   device(config-pki-trustpoint-trust1)# auto-enroll
   device(config-pki-trustpoint-trust1)# enrollment retry-period 2
   device(config-pki-trustpoint-trust1)# enrollment profile profile1
   device(config-pki-trustpoint-trust1)# pki-entity entity1
   device(config-pki-trustpoint-trust1)# eckeypair key-label eckeyAuto
   device(config-pki-trustpoint-trust1)# fingerprint 36:0c:92:6e:df:b2:72:eb:59:e8:63:73:2a:98:a8:91:cb:
   50:94:d9
   device(config-pki-trustpoint-trust1)# ocsp http post
   device(config-pki-trustpoint-trust1)# exit
   ```

6. Authenticate the CA (trust1 in this example) to the FastIron device by obtaining the self-signed certificate from the CA.

   The following example authenticates previously configured trustpoint trust1.

   ```
   device(config)# pki authenticate trust1
   ```

7. Once the trustpoint has been authenticated, enroll the FastIron device with the PKI trustpoint to obtain a local certificate signed by the CA server. The **pki enroll** command sends a CSR request to the CA with the configured keypair and entity values. The CA server signs it and sends back the client certificate for the given trustpoint.

   The following example enrolls the PKI trustpoint trust1.

   ```
   device(config)# pki enroll trust1
   ```

   The FastIron device, once enrolled, requests certificates from the CA for each of its key pairs. The CA sends the response in the form of a local certificate.

The following example creates a PKI entity, configures a PKI enrollment profile, and specifies automatic enrollment as the enrollment method. It then configures an enrollment profile. Next, it creates a trustpoint containing the previously configured enrollment profile and PKI entity. Finally, it authenticates and enrolls the trustpoint.

```
device# configure terminal
device (config)# pki entity entity1
device(config-pki-entity-entity1)# common-name "tester1"
device(config-pki-entity-entity1)# country-name "IN"
device(config-pki-entity-entity1)# state-name "KA"
device(config-pki-entity-entity1)# org-unit-name "FI"
device(config-pki-entity-entity1)# org-name "Ruckus"
device(config-pki-entity-entity1)# email-id "user@ruckus.com"
device(config-pki-entity-entity1)# location "BG"
device(config-pki-entity-entity1)# exit
device(config)# pki profile-enrollment profile1
device(config-pki-profile-enrollment-profile1)# authentication-url http://WIN-N6C3R0LUDAJ.englab.ruckus.com/
CertSrv/mscep/mscep.dll
device(config-pki-profile-enrollment-profile1)# authentication-command WIN-
N6C3R0LUDAJ.englab.ruckus.com_englab-WIN-N6C3R0LUDAJ-CA-15
device(config-pki-profile-enrollment-profile1)# enrollment-url http://WIN-N6C3R0LUDAJ.englab.ruckus.com/
CertSrv/mscep/mscep.dll
device(config-pki-profile-enrollment-profile1)# password DB6E1F091AEF0244
device(config-pki-profile-enrollment-profile1)# exit
device(config)# pki trustpoint trust1
device(config-pki-trustpoint-trust1)# auto-enroll
device(config-pki-trustpoint-trust1)# enrollment retry-period 2
device(config-pki-trustpoint-trust1)# enrollment profile profile1
device(config-pki-trustpoint-trust1)# pki-entity entity1
device(config-pki-trustpoint-trust1)# eckeypair key-label eckeyAuto
device(config-pki-trustpoint-trust1)# fingerprint 36:0c:92:6e:df:b2:72:eb:59:e8:63:73:2a:98:a8:91:cb:
50:94:d9
device(config-pki-trustpoint-trust1)# ocsp http post
device(config-pki-trustpoint-trust1)# exit
device(config)# pki authenticate trust1
device(config)# pki enroll trust1
```

PKI Manual Import

To import X.509 certificates, you need the following items:

1. Root CA Certificate, which will be used with the configured trustpoint on the TOE.
2. The intermediate CA certificate.
3. The local TOE certificate, which is signed by the RootCA or any intermediate CA of the RootCA.
4. The key for the TOE's local certificate.

The listed certificates and keys should be copied into flash on the TOE.

NOTE
Refer to Using SCP to copy a digital certificate to a FastIron device on page 27.

Perform the following steps to import the certificates manually.

1. Create a trustpoint. Here, the trustpoint corresponds to ROOT CA. Set the fingerprint for the trustpoint (which can be copied from the rootCA certificate).

   The following example provides the syntax for the commands.

   ```
   device# configure terminal
   device(config)# pki trustpoint < trustpointname >
   device(config-pki-trustpointname)# fingerprint < value >
   ```

   The following example creates a trustpoint called abcd and sets the fingerprint to the value copied from the rootCA certificate.

   ```
   device# configure terminal
   device(config)# pki trustpoint abcd
   device(config-pki-trustpoint-abcd)# fingerprint 3C:EA:EC:E6:F1:DD:3B:86:65:DE:58:F4:A2:75:D8:63:6D:
   23:68:40
   device(config-pki-trustpoint-abcd)# exit
   ```

2. Import the key to the PKI database from flash memory of the FastIron device using one of the following command.

   NOTE
   The key file can be a plain text or an encrypted file (encryption is recommended). Only aes256 encryption is supported for key file encryption.
   The following examples provide command syntax. The key type may be either **rsa** or **ec**.
   If the key file being used is plain text, the following command format should be used:

   ```
   device(config)# pki import key < keytype > < keylabel > pem url flash: < keyname >.key.pem
   ```

   If the key file is encrypted using a password, the following command format should be used:

   ```
   device(config)# pki import key < keytype > < keylabel > pem url flash: < keyname >.key.pem
   password pwd_value
   ```

   The following example imports the RSA key from the flash location specified.

   ```
   device(config)# pki import key rsa rsakey pem url flash: dut.key.pem
   device(config)# end
   ```

   The following example imports the EC key eckey1 from flash using a password string associated with the encrypted file. (You will be prompted for the password if you later try to open the imported file.)

   ```
   device(config)# pki import key ec eckey1 pem url flash: dut.eckey.pem password f1234yzztk!
   ```

3. Use the following commands to import the CA and local certificates.

   The following example provides command syntax.

   ```
   device(config)# pki import < trustpointname > pem url flash: rootca.pem
   device(config)# pki import < trustpointname > pem url flash: localcert.pem
   ```

   The following command imports the CA certificate (rootca.pem) and the local certificate (localcert.pem) from the trustpoint named abcd.

   ```
   device(config)# pki import abcd pem url flash: rootca.pem
   device(config)# pki import abcd pem url flash: localcert.pem
   ```

4. Attach the imported key label to the trustpoint using the following commands.

   The following example shows command syntax.

   ```
   device(config)# pki trustpoint < trustpointname >
   device(config-pki-trustpoint-trustpointname)# { rsakeypair | eckeypair } key-label <label >
   ```

   The following example binds the rsakey label to the CA trustpoint named abcd.

   ```
   device(config)# pki trustpoint abcd
   device(config-pki-trustpoint-abcd)# rsakeypair key-label rsakey
   ```

5. Authenticate the rootCA trustpoint using the following command.

   The following example provides the syntax for authenticating the trustpoint.

   ```
   device(config-pki-trustpoint-trustpointname)# exit
   device(config)# pki authenticate < trustpointname >
   ```

   The following example authenticates the previously configured trustpoint abcd.

   ```
   device(config-pki-trustpoint-abcd)# exit
   device(config)# pki authenticate abcd
   ```

   NOTE

   Because the manual process does not generate a CSR on the FastIron device (the TOE), it is not necessary to execute the **pki enroll** command. after the trustpoint is authenticated.

6. Use the following commands to check the local and CA certificates on the FastIron device.

```
device(config)# show pki certificates local
device(config)# show pki certificates trustpoint < trustpointname >
```

The following example displays information for the local certificate and for certificates from the trustpoint abcd.

```
device# show pki certificates local

----------------PKI LOCAL CERTIFICATE ENTRY-----------------
CA: TLS-ABCD
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4125 (0x101d)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com
        Validity
            Not Before: Nov  7 02:24:18 2017 GMT
            Not After : Nov 17 02:24:18 2018 GMT
        Subject: CN=DUTFIPSCC, ST=CA, C=US/emailAddress=user@arris.com, O=DUTFIPSCC, OU=SQA
device#


device# show pki certificates trustpoint

----------------PKI TRUSTPOINT CERTIFICATE ENTRY-----------------
CA: TLS-ABCD
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            bd:fa:4f:da:bd:89:4a:5d
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com
        Validity
            Not Before: Nov  7 02:10:00 2017 GMT
            Not After : Nov  7 02:10:00 2022 GMT
        Subject: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com
device#
```

7. Validate the certificates using the following command.

The following example provides the syntax for validating the imported certificates.

```
device(config)# pki cert-validate < trustpointname >
```

The following example successfully validates certificates imported from the trustpoint abcd.

```
device(config)# pki cert-validate abcd
PKI: Successfully validated the local certificate for trustpoint: abcd
```

The following example imports certificates for the configured trustpoint abcd, authenticates the trustpoint, and validates imported certificates.

```
device# configure terminal
device(config)# crypto key generate rsa
device(config)# pki import key rsa rsakey pem url flash: dut.key.pem
device(config)# pki import abcd pem url flash: rootca.pem
device(config)# pki import abcd pem url flash: localcert.pem
device(config)# pki trustpoint abcd
device(config-pki-trustpoint-abcd)# ocsp http post
device(config-pki-trustpoint-abcd)# revocation-check ocsp
device(config-pki-trustpoint-abcd)# ocsp-url http://10.21.40.39:2560
device(config-pki-trustpoint-abcd)# fingerprint 3C:EA:EC:E6:F1:DD:3B:86:65:DE:58:F4:A2:75:D8:63:6D:23:68:40
device(config-pki-trustpoint-abcd)# exit
device(config)# pki authenticate abcd
device(config)# pki cert-validate abcd
PKI: Successfully validated the local certificate for trustpoint: abcd
```

Revocation Check for Peer Certificates

FastIron devices in Common Criteria mode support OCSP for checking the revocation status of a certificate received from a peer. The revocation status is checked for both the intermediate and the peer certificate. The following actions are taken in revocation checks:

- The OCSP signing bit should be set for the OCSP responder.
- If the OCSP responder responds that the certificate status is good, the TLS session comes up.
- If the OCSP responder responds that the certificate status is revoked, the TLS session does not come up.
- If the OCSP server is not reachable and the response is not received, then the TLS session does not come up between the peers.

The OCSP URL is picked up from the certificates received from the peer. The default configuration is "revocation-check none" and the status of the revoked certificate is not checked. When the command **revocation-check ocsp** is configured as part of the trustpoint configuration, OCSP protocol is used to check the status of certificates received from the peer. The command "ocsp-url http://15.1.1.1:2560" is used to identify the URL for the OCSP responder.

By default, an HTTP "get" command is used to reach the OCSP responder. This method can be changed to an HTTP post using the command "ocsp http post" as shown in the following example.

```
device# configure terminal
device(config)# pki trustpoint abcd
device(config-pki-trustpoint-abcd)# ocsp http post    <-- Optional command. Applies only to Linux.
device(config-pki-trustpoint-abcd)# revocation-check ocsp
device(config-pki-trustpoint-abcd)# ocsp-url http://15.1.1.1:2560
device(config-pki-trustpoint-abcd)# rsakeypair key-label rsakey
device(config-pki-trustpoint-abcd)# fingerprint 3C:EA:EC:E6:F1:DD:3B:86:65:DE:58:F4:A2:75:D8:63:6D:23:68:40
device(config-pki-trustpoint-abcd)# exit
device(config)#
```

Network Device Collaborative Protection Profile

The Network Collaborative Device Protection Profile (NDcPP) standards provide a set of rules that define the security requirements for network devices. The main purpose of these requirements is to minimize and reduce threats to network devices. NDcPP requires SSH or TLS for syslog and authentication server communications.

> **NOTE**
> The connections syslog and AAA servers need to be over TLS. For more information, see the output of **fips enable common-criteria** command with lines prefixed with "CC".

Audit Logging

FastIron devices support logging of PKI transaction details. The logs are automatically generated syslog messages that contain the PKI transaction details.

There are two types or levels of logging. Standard logging is enabled by default. The second type of logging is called extended logging, which you must enable using commands. This type of logging allows you to log additional PKI transaction details.

Support for Logging PKI Transaction Details

FastIron devices support logging of PKI transaction details. The log files are automatically generated syslog messages that contain the transaction details.

There are two types or levels of logging. Standard logging is enabled by default. The second type of logging is called extended logging, which you must enable using commands. This type of logging allows you to log additional IKE or PKI transaction details.

The following additional logging options can be configured in general configuration mode:

- **logging enable pki**
- **logging enable pki pki-extended**

For example, enter the following commands to configure extended logging for PKI:

```
device# configure terminal
device(config)# logging enable pki-extended
```

Once the extended logging commands are configured for PKI, the logs listed in the following table are generated on the ICX device.

TABLE 7 Extended Logging Messages

Event	Audit Log
Certificate time (validity period) expired.	Certificate has expired.
Signature is not valid.	Certificate signature failure.
Extended Key Usage support does not have expected key purposes.	Unsupported certificate purpose.
Certificate is revoked by CA (applies to both chain and non-chained case).	Revoked.
Wrong root certificate received in a certificate chain.	Unable to get local issuer certificate.
Configured DN value does not match with peer certificate remote DN.	Hostname mismatch.
OCSP Response does not have OCSPSigning bit set.	OCSP purpose missing in responder certificate.
There is a fingerprint mismatch.	Fingerprint match failed.

Limitations

All of the current limitations of the logging feature on FastIron devices apply to the logging of PKI transaction details.

Management Commands

The following list of commands and command variants are required for administration of the TOE. These commands are available only after an administrator has successfully logged into the TOE.

TABLE 8 Management Commands

Command	Tested Command Variants	Description
aaa	**aaa** authentication **aaa** authentication enable default radius local **aaa** authentication login default radius local **aaa** authentication web-server default local	Configures the AAA authentication functions
banner	**banner** motd+	Manages the login banner
clock	**clock** *set time*	Manages the internal clock
config	**config** terminal	Switches to configuration mode
crypto	**crypto key generate**	Invokes cryptographic functions.
crypto-ssl	**crypto-ssl** certificate generate	Manages web server properties.
exit	**exit**	Logs out or exits current session. Used to terminate both local console and remote SSH sessions.
fips	**fips** enable common-criteria **fips** show **fips** zeroize all	Manages FIPS and common criteria configuration.

TABLE 8 Management Commands (continued)

Command	Tested Command Variants	Description
interface	**interface** ethernet 4/12 **interface** mac access-group 400 in **tunnel**	Configures an interface or associates an ACL with an interface.
ip / ipv6	**access-list** **access-group** **address** **ssl profile** (IPv4 only)	Configures IPv4 and IPv6 parameters.
logging	**logging host** *ip-address* **ssl-port** *port-number* **profile** *profile-name*	Configures the audit logging host.
logging enable	**logging enable pki**	Configures PKI logging.
	logging enable pki pki-extended	Configures PKI extended logging.
openssl	**openssl s_server**	Configures secure connections (for example with syslog).
pki	**pki** **authenticate** - Authenticates CA to router by obtaining the self-signed certificate of the CA. **cert-validate** - Determines if a trustpoint has been successfully authenticated. **enroll** - Requests certificates from the CA for each key pair of your router. **entity** - Configures PKI end-user parameters. **export** - Exports a PKI certificate manually. **import** - Imports a PKI certificate manually. **profile-enrollment** - Configures PKI enrollment parameters. **trustpoint** - Configures PKI CA parameters.	Configures Public Key Infrastructure parameters.
radius-server	**radius-server host** *ip-address* **ssl-auth-port** *port* **profile** *profile-name* **authentication key** *value* **radius-server** retransmit *retransmit period* **radius-server** timeout *timeout period* **radius-server** key *key name*	Configures the RADIUS server.
reload	**reload**	Reloads the current flash image.
server	**server** *ntp server ip* **minpoll** *time*	Configures external services.
show	**show** flash **show** version **show** clock **show** ip client-pub-key **show** ip ssl **show** logging **show** pki **show** running-config	Displays information about specified configuration.

TABLE 8 Management Commands (continued)

Command	Tested Command Variants	Description
timeout	**cli timeout** *time*	Configures the timeout for CLI inactivity in minutes. The default is 2 minutes. The CLI session timeout also controls SSH sessions. Valid values are 0-240. **NOTE** It is recommended that you configure a non-zero value so that a timeout occurs any time the system is idle.
username	**username** *user* password	Manages user accounts.
write	**write memory**	Writes to persistent storage.

MACsec Configuration

MACsec Overview

Media Access Control Security (MACsec) is a Layer 2 security technology that provides point-to-point security on Ethernet links between nodes.

NOTE
MACsec is supported on RUCKUS ICX 7450, ICX 7550, ICX 7650, and ICX 7850 devices.

MACsec, defined in the IEEE 802.1AE-2006 standard, is based on symmetric cryptographic keys. MACsec Key Agreement (MKA) protocol, defined as part of the IEEE 802.1x-2010 standard, operates at Layer 2 to generate and distribute the cryptographic keys used by the MACsec functionality installed in the hardware.

As a hop-to-hop Layer 2 security feature, MACsec can be combined with Layer 3 security technologies such as IPsec for end-to-end data security.

Supported MACsec Hardware Configurations

MACsec key-enabled security can be deployed on a point-to-point LAN between two connected ICX devices over interfaces that share a preconfigured static key, the Connectivity Association Key (CAK).

On a licensed ICX 7450, ICX 7550, ICX 7650, or ICX 7850 device, 10-Gbps ports can be configured for MACsec. Licenses are available per device as described in the *RUCKUS FastIron Software Licensing Guide*.

NOTE

1. On ICX 7450 devices, MACsec is available only on 4 X 10GF modules installed in slots 2, 3, or 4.
2. On ICX 7550 devices, MACsec is available only on 4 X 10GF modules installed in slot 3.
3. On ICX 7650 devices, MACsec is available only on 10-Gbps fiber ports, that is, ports 25 through 48 of the base module for ICX 7650-48F devices or on slot 2 when a 4 X 10GF module is installed.
4. MACsec is available on 10-Gbps ports of ICX 7850-48FS devices only.

MACsec RFCs and Standards

FastIron MACsec complies with the following industry standards:

- IEEE Std 802.1X-2010: Port-Based Network Access Control
- IEEE Std 802.1AE-2006: Media Access Control (MAC) Security

- RFC 3394: Advanced Encryption Standard (AES) Key Wrap Algorithm
- RFC 5649: Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm

MACsec Considerations

Review the following considerations before deploying MACsec:

- As a prerequisite, MACsec must be licensed on each device where it is used.
- MACsec introduces an additional transit delay, due to the increase in the MAC Service Data Unit (MSDU) size.
- MACsec and Flexible authentication cannot be configured on the same port.
- On an ICX 7450 device or an ICX 7550 device, ports on a 4 X 10GF removable module installed in ICX 7450 slot 2 or ICX 7550 slot 3 can be used for MACsec or stacking but not both simultaneously.
- In rear modules 3 and 4 on an ICX 7450 device, MACsec can be supported at all times because stacking is not available on those modules.

Configuring MACsec

Although the MACsec configuration options outlined in this section are always visible, they cannot be applied unless an active license is present on the device and MACsec is enabled. MACsec licenses are required on a per-device basis. Each device in a stack requires a separate MACsec license.

These steps are required to configure MACsec security on a link or a group of connected ports.

1. Enter the dot1x-mka level from the global configuration level, and enable MACsec for the device.
2. Configure the MACsec Key Agreement (MKA) group.
3. Configure required parameters for the group, including frame validation, confidentiality, replay protection or delay protection, and actions to be taken when MACsec requirements are not met.
4. Create and configure an MKA keychain.
5. Enable MKA on each participating interface.
6. Apply the configured MKA group on the participating interface.

   NOTE
   If an MKA group is not applied to an enabled MACsec interface, or if parameters within the applied group have not been configured, default values are applied to the interface. Configured parameters are visible in **show** command output; default parameters are not always visible. Refer to the *RUCKUS FastIron Command Reference Guide* for default values for each command.

7. Apply the previously configured MKA keychain to the MACsec interface.

   NOTE
   As an alternative, you can configure the Connectivity Association Key (CAK) and Connectivity Association Key Name (CKN) on each interface. However, pre-shared key configuration and MKA keychain configuration are not allowed on the same interface.

Enabling MACsec and Configuring Group Parameters

Enable MACsec globally on the device, and configure the MACsec Key Agreement (MKA) group before configuring MACsec security features for the group.

A valid license must be installed on the device before MACsec can be configured.

1. At the global configuration level, enter the **dot1x-mka-enable** command to enable MACsec on the device.

```
device# configure terminal
device(config)# dot1x-mka-enable
device(config-dot1x-mka)#
```

   MACsec is enabled, and the device is placed at the dot1x-mka configuration level.

   NOTE
   When MKA is disabled, all the ports are brought to a down state. You must manually enable the ports again to bring the ports back up.

2. Enter the **mka-cfg-group** command followed by a group name to create a group.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)#
```

   The group is created, and the device is placed at the group configuration level.

Next: At the group configuration level, set the key-server priority, and define MACsec security features to be applied to interfaces.

Configuring MACsec Key-Server Priority

MACsec uses a key-server to generate and distribute encryption parameters and secure key information to members of a MACsec connectivity association.

The key-server is elected by comparing key-server priority values during MACsec Key Agreement (MKA) message exchange between peer devices. The elected key-server is the peer with the lowest configured key-server priority, or with the lowest Secure Channel Identifier (SCI) in case of a tie. Key-server priority may be set to a value from 0 through 255. When no priority is configured, the device defaults to a priority of 16, which is not displayed in MACsec configuration details.

NOTE
If the key-server priority is set to 255, the device will not become the key-server.

Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

At the dot1x-mka group configuration level, enter the **key-server-priority** command, and specify a value from 0 through 255 to define the key-server priority.

```
device# configure terminal
device(config)# dot1x-mka-enable
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# key-server-priority 20
```

In this example, the key-server priority is set to 20 for the MKA group test1.

Next: Configure MACsec integrity and encryption for the group.

Configuring MACsec Integrity and Encryption

To ensure point-to-point integrity, MACsec computes an Integrity Check Value (ICV) on the entire Ethernet frame using the designated cipher suite. The designated cipher suite is also used for encryption.

MACsec adds the ICV to the frame before transmission. The receiving device recalculates the ICV and checks it against the computed value that has been added to the frame. Because the ICV is computed on the entire Ethernet frame, any modifications to the frame can be easily recognized.

By default, both encryption and integrity protection are enabled.

MACsec encrypts traffic between devices at the MAC layer and decrypts frames within participating networked devices. MACsec uses the Galois/Counter Mode Advanced Encryption Standard 128 or 256 (GCM-AES-128 or GCM-AES 256) cipher suite to encrypt data and to compute the ICV for each transmitted and received MACsec frame.

MACsec also encrypts the VLAN tag and the original Ethertype field in the Layer 2 header of the secured data. When initial bytes in a secure data packet must be transparent, a confidentiality offset of 30 or 50 bytes can be applied.

NOTE
Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

1. At the dot1x-mka group configuration level, enter the **macsec cipher-suite** command with one of the available options:
   - **gcm-aes-128**: Enables encryption and integrity checking using the GCM-AES-128 cipher suite.
   - **gcm-aes-128 integrity-only**: Enables integrity checking without encryption.
   - **gcm-aes-256**: Enables encryption and integrity checking using the GCM-AES-256 cipher suite.

     NOTE
     The **gcm-aes-256** option is not supported on ICX 7450 devices.

   - **gcm-aes-256 integrity-only**: Enables integrity checking without encryption.

   In the following example, MACsec 128-bit encryption has been configured as a group test1 setting. By default, the ICV integrity check is also enabled, no matter which cipher suite you use.

   ```
   device# configure terminal
   device(config)# dot1x-mka
   device(config-dot1x-mka)# mka-cfg-group test1
   device(config-dot1x-mka-group-test1)# macsec cipher-suite gcm-aes-128
   ```

   In the following example, MACsec has been configured for integrity protection only, without encryption.

   ```
   device# configure terminal
   device(config)# dot1x-mka
   device(config-dot1x-mka)# mka-cfg-group test1
   device(config-dot1x-mka-group-test1)# macsec cipher-suite gcm-aes-128 integrity-only
   ```

   NOTE
   The **no macsec cipher-suite** command disables both encryption and integrity checking.

2. Enter the **macsec confidentiality-offset** command if an encryption offset is required:
   - **30**: Encryption begins at byte 31 of the data packet.
   - **50**: Encryption begins at byte 51 of the data packet.

   NOTE
   The default offset for MACsec encryption is zero bytes. Use the **no macsec confidentiality-offset** command to return the offset to zero bytes.

   In the following example, the encryption offset is defined as 30 bytes. The first 30 bytes of each data packet carried within the MACsec frame are transmitted without encryption.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# macsec confidentiality-offset 30
```

Configuring MACsec Frame Validation

You can specify whether incoming frames are checked for MACsec (secTAG) headers and how invalid frames are handled.

NOTE
Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

At the MKA group configuration level, enter the **macsec frame-validation** command, and select an option:

- **disable**: Received frames are not checked for a MACsec header.
- **check**: If frame validation fails, counters are incremented, but packets are accepted.
- **strict**: If frame validation fails, packets are dropped, and counters are incremented.

In the following example, group test1 is configured to validate frames and discard invalid ones.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# macsec frame-validation strict
```

Configuring Replay Protection

MACsec replay protection detects repeated or out-of-order packets and acts as a safeguard against man-in-the-middle attacks.

When replay protection is configured, MACsec uses a separate replay packet number (PN) counter and gives each Ethernet frame a packet number. As frames are received, packet numbers are monitored.

Two modes of replay protection are supported: strict and out-of-order. In strict mode (the default), packets must be received in the correct incremental sequence. In out-of-order mode, packets are allowed to arrive out of sequence within a defined window.

NOTE
Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

At the dot1x-mka group configuration level, enter the **macsec replay-protection** command with one of the available modes:

- **strict**: Frames must be received in exact incremental sequence.
- **out-of-order** *window size*: Frames are accepted out of order within the designated window size. The maximum window size is 4294967295.
- **disable**: Frames are not validated.

In the following example, MACsec replay protection is enabled for group test1. Frames must be received in exact order.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# macsec replay-protection strict
```

In the following example, MACsec replay protection is enabled for group test1. Frames are accepted out of order within the designated window size (100).

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# macsec replay-protection out-of-order window-size 100
```

Once you have configured the desired MKA group settings, these settings can be applied to specific interfaces.

Configuring Data-Delay Protection

MACsec data-delay protection allows MKA participants to ensure that the data frames protected by MACsec are not delayed by more than two seconds.

Each MACsec peer uses the MACsec Key Agreement (MKA) Protocol Data Unit (MKPDU) to communicate the lowest acceptable packet number. When a peer receives MACsec data with a packet number value less than the lowest acceptable packet number, MACsec increments the Delay Packet counters.

By default, the data-delay protection feature is disabled. Configuring the **macsec delay-protection** command under MKA group settings and attaching the group to a MACsec interface enables the data-delay protection feature on that interface.

NOTE
MACsec replay protection must be disabled when MACsec data-delay protection is configured.

NOTE
MACsec data-delay protection is not available on ICX 7450 devices.

NOTE
Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

At the dot1x-mka group configuration level, enter the **macsec delay-protection** command.

In the following example, data-delay protection is enabled for group test1. Frames are protected when they are received with a delay of two seconds or less.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka)# macsec replay-protection disable
device(config-dot1x-mka-group-test1)# macsec delay-protection
```

Once you have configured the desired MKA group settings, these settings can be applied to specific interfaces.

MKA Keychain Overview and Considerations

Keychains are an alternative to configuring a single pre-shared key on each MACsec interface. You can configure a maximum of eight MKA keychains, and each keychain can contain a maximum of 32 configured keys. Each key contains a pre-configured password, authentication algorithm, and a send lifetime configuration.

Keep the following considerations in mind when configuring MKA keychains.

- When you use the keychain feature for a MACsec interface, the authentication algorithm configured under **mka-keychain** is used. The authentication algorithm configured under **mka-cfg-group** is not used.
- An MKA keychain cannot be modified after it is bound to an interface.
- An MKA keychain does not support a combination of AES-128 and AES-256 algorithms. Only one of the two algorithms can be configured in the same keychain.
- The 'accept-lifetime' configuration under the keychain module configuration does not apply to the MACsec keychain.
- The interval between Start and End lifetime for a configured MKA key must be a minimum of 120 seconds.
- The interval between activation of two successive MKA keys must be a minimum of 120 seconds.
- All invalid keys in the MKA keychain will be ignored.

Creating and Configuring an MKA Keychain

Perform the following steps to configure an MKA keychain.

1. In global configuration mode, enter the **keychain** command followed by the keychain name and the keyword **mka**.

```
device# configure terminal
device(config)# keychain sample mka
device(config-keychain-mka-sample)#
```

   The keychain is created, and the device is placed in MKA keychain configuration mode.

2. Configure the key identifier. Valid values are from 1 through 4294967296.

```
device(config-keychain-mka-sample)# key-id 100
```

   The key is created, and the device is placed in MKA key configuration mode.

3. Configure the authentication algorithm for the key. Options are **authentication-algorithm aes-128-cmac** and **authentication-algorithm aes-256-cmac**.

   NOTE
   ICX 7450 devices support only AES-128-CMAC.

```
device(config-keychain-mka-sample-key-100)# authentication-algorithm aes-128-cmac
```

4. Configure the password for the key.

   NOTE
   Passwords are composed of hexadecimal characters 0 through 9 and a through f. When AES-128-CMAC is used as the authentication algorithm, 16 hexadecimal characters must be configured. When AES-256-CMAC is used, 32 hexadecimal characters must be configured.

```
device(config-keychain-mka-sample-key-100)# password
12345678123456781234567809abcdef12345678123456781234567809abcdef
```

5. Configure the send-lifetime start and end times.

   NOTE
   If you use the keywords **end infinite** as shown in the example instead of a specific end time, the key remains active indefinitely.

```
device(config-keychain-mka-sample-key-100)# send-lifetime start 02-14-2022 01:01:01 end infinite
```

6. (Optional) Configure the local timezone (as configured in the system) to be used for the start and end timers. If not configured, the lifetime values are based on the GMT clock time.

```
device(config-keychain-mka-sample-key-100)# send life-time local
```

7. (Optional) Configure the tolerance value for the keys.

NOTE
Because of the potential for key overlap when the duration between the first key end-time and the following key start-time is short, RUCKUS recommends that you configure a minimum tolerance of 180 seconds to maintain hitless key rollover.

The following example configures a 200 second tolerance period for the keychain profile "mka-sample-key-100."

```
device(config-keychain-mka-sample-key-100)# tolerance 200
```

The following example creates the MKA keychain "sample" and configures the underlying options. The configured options are confirmed in the output of the **show keychain name** command.

```
device# configure terminal
device(config)# keychain sample mka
device(config-keychain-mka-sample)# key-id 100
device(config-keychain-mka-sample-key-100)# authentication-algorithm aes-128-cmac
device(config-keychain-mka-sample-key-100)# password
12345678123456781234567809abcdef12345678123456781234567809abcdef
device(config-keychain-mka-sample-key-100)# send-lifetime start 02-16-2022 04:05:00 end infinite
device(config-keychain-mka-sample-key-100)# tolerance 200
device(config-keychain-mka-sample-key-100)# end
device# show keychain name sample
Keychain: sample
  Tolerance: 0
  Key-id   : 100
      AuthAlgorithm: aes-128-cmac
      Key-String   : *******
      Send Lifetime:-
          Start    : 02-16-2022 04:05:00 End         : Infinite
          Active   : Yes                 TimeToExpire: Infinite
          Timezone : GMT+00
```

Next: Enable interfaces for MACsec, and apply the MKA group configuration and MKA keychain as described in Enabling and Configuring Group Interfaces for MACsec on page 67.

Enabling and Configuring Group Interfaces for MACsec

After MACsec is enabled for the device, each MACsec interface must be individually enabled, and a configured set of parameters must be applied.

1. To enable MACsec, at the dot1x-mka configuration level, enter the **enable-mka** command, and specify the interface as *unit/slot/port*.

In the following example, Ethernet port 1 on slot 2 of unit 2 is enabled for MACsec security.

```
device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# enable-mka ethernet 2/2/1
device(config-dot1x-mka-2/2/1)#
```

NOTE
The following output is displayed if there is no MACsec license present on the device.

```
device(config-dot1x-mka)# enable-mka ethernet 2/2/1
Error: No MACsec License available for the port 2/2/1. Cannot enable MACsec !!!
Error: MKA cannot be enabled on port 2/2/1
```

2. At the dot1x-mka interface configuration level, enter the **mka-cfg-group** command, and specify the MKA group configuration to apply to the interface.

```
device(config-dot1x-mka-2/2/1)# mka-cfg-group test1
```

3. **NOTE**
   This step is an option to configuring the pre-shared key as described later. For more information on MKA keychains and how to configure them, refer to MKA Keychain Overview and Considerations on page 65.

   At the dot1x-mka interface configuration level, enter the **mka-keychain** command, and specify the previously configured keychain to apply to the interface.

```
device(config-dot1x-mka-2/2/1)# mka-keychain macsec1
```

In the following example, MACsec options configured for "group test1" and MKA keychain "macsec1" are applied to the enabled interface.

```
device# configure terminal
device(config)# dot1x-mka
device (config-dot1x-mka)# enable-mka ethernet 2/2/1
device(config-dot1x-mka-2/2/1)# mka-cfg-group test1
device(config-dot1x-mka-2/2/1)# mka-keychain macsec1
device(config-dot1x-mka-2/2/1)# end
device#
```

Configuring the Pre-shared Key

NOTE
Refer to Configuring MACsec on page 61 for an overview of enabling and configuring MACsec features.

MACsec security is based on a pre-shared key, the Connectivity Association Key (CAK), which you define and name. Only MACsec-enabled interfaces that are configured with the same key can communicate over secure MACsec channels. The key can be configured directly on each MACsec interface, or you can configure a set of keys as an MKA keychain and apply the keychain to each interface. MKA keychain configuration is described in Creating and Configuring an MKA Keychain on page 66.

NOTE
Pre-shared key configuration and 'mka-keychain' configuration are not allowed on the same interface. If you have already configured and applied an MKA keychain to the MACsec interface, this task is not required.

At the dot1x-mka-interface configuration level, enter the **pre-shared-key** command followed by the *key-id*, the keyword **key-name**, and a hex string to define and name the pre-shared key.

The requirements for the parameters are as follows:

- *key-id*: Define the key ID value using 32 hexadecimal characters.
- key-name *hex string*: Give the key a name using from 2 through 64 hexadecimal characters (in 8-bit multiples).

In the following example, the pre-shared key with the hex value beginning with "135bd758b" and the key name beginning with "96437a93" are applied to interface 1/3/2.

```
device# configure terminal
device(config)# dot1x-mka
device (config-dot1x-mka)# enable-mka ethernet 1/3/2
device(config-dot1x-mka-1/3/2)# pre-shared-key 135bd758b0ee5c11c55ff6ab19fdb199 key-name
96437a93ccf10d9dfe347846cce52c7d
```

Enable and configure each MACsec interface. Configure the same pre-shared key (CAK) on the interfaces between which a secure channel can be established.

Sample MACsec Configuration

The following example shows how to enable MACsec, configure general parameters, enable and configure interfaces, and create and assign a keychain module to an interface. The keychain module must also be assigned to peer interfaces.

```
device# configure terminal
device(config)# dot1x-mka-enable

device(config-dot1x-mka)# mka-cfg-group test1
device(config-dot1x-mka-group-test1)# key-server-priority 5
device(config-dot1x-mka-group-test1)# macsec cipher-suite gcm-aes-128
device(config-dot1x-mka-group-test1)# macsec confidentiality-offset 30
device(config-dot1x-mka-group-test1)# macsec frame-validation strict
device(config-dot1x-mka-group-test1)# macsec replay-protection strict
device(config-dot1x-mka-group-test1)# exit
device(config-dot1x-mka)# exit

device(config)# keychain macsec1 mka
device(config-keychain-mka-macsec1)# key-id 1
device(config-keychain-mka-macsec1-key-1)# password ........
device(config-keychain-mka-macsec1-key-1)# authentication-algorithm aes-256-cmac
device(config-keychain-mka-macsec1-key-1)# send-lifetime start 02-14-2022 01:01:01 end 03-14-2022 06:59:00
device(config-keychain-mka-macsec1-key-1)# end

device# configure terminal
device(config)# dot1x-mka
device(config-dot1x-mka)# enable-mka ethernet 1/3/2

device(config-dot1x-mka-1/3/2)# mka-cfg-group test1
device(config-dot1x-mka-1/3/2)# mka-keychain macsec1
device(config-dot1x-mka-1/3/2)# end
device#
```

Displaying MACsec Information

Use MACsec **show** commands to display information on MACsec for a device, group, or individual interface. MACsec **show** commands can be used to display configuration information, to report on MACsec sessions that are currently active on a device, or to monitor MACsec statistics on a particular interface.

Displaying MACsec Configuration Details

You can display configuration information for all MACsec groups on a device, or you can display details for a particular group.

1. In privileged EXEC, global configuration, or dot1x-mka interface mode, use the **show dot1x-mka config** command to display MACsec configuration details for the device.

   In the following example, MACsec parameters are displayed for the device and all groups configured on it. Specific MACsec interfaces are displayed as well as the pre-shared key for each interface.

   NOTE
   The pre-shared key is displayed as an encrypted value, not in plain text.

```
device(config)# show dot1x-mka config
dot1x-mka-enable
mka-cfg-group group1
  key-server-priority 20
  macsec frame-validation check
  macsec confidentiality-offset 30
  macsec cipher-suite gcm-aes-128
  macsec-replay protection out-of-order window-size 100
  enable-mka ethernet 1/3/2
mka-cfg-group group1
  pre-shared-key xxxxxxxxxxxxxxxxxxxxxxxx key-name 96437a93ccf10d9dfe3478460cce5132
enable-mka ethernet 1/3/6
  mka-cfg-group group1
  pre-shared-key xxxxxxxxxxxxxxxxxxxxxxxx key-name 96437a93ccf10d9dfe3478460cce51321
```

2. In privileged EXEC, global configuration, or dot1x-mka interface mode, enter the **show dot1x-mka config-group** command to display information for all configured groups. Add a group name to the command to narrow the information displayed to one group.

   The following example displays information for MKA group test1.

```
device(config)# show dot1x-mka config-group test1
 mka-cfg-group test1
  key-server-priority 5
  macsec cipher-suite gcm-aes-128 integrity-only
  macsec confidentiality-offset 30
  macsec frame-validation strict
  macsec replay-protection strict
```

   NOTE
   Group information does not include the pre-shared key or enabled connections. Use the **show dot1x-mka config** command to obtain that information.

3. In privileged EXEC, global configuration, or dot1x-mka interface mode, use the **show keychain mka** command to display details for configured MACsec keychains. For additional details on a specific keychain, use the **show keychain name** command.

   The following example shows that one MKA keychain named "sample" has been configured.

```
device# show keychain mka

Keychain  : sample
Tolerance : 0
------------------------------------------------------------------------------
  Key-id  |    Algo    | SendActive | SendTimer | AcceptActive | AcceptTimer
------------------------------------------------------------------------------
 100       aes-128-cmac Yes(GMT+00)      -        No(GMT+00)          -
```

   The following example uses the **show keychain name** command followed by the keychain name displayed in the previous example to display additional details for the specified keychain.

```
device# show keychain name sample
Keychain: sample
  Tolerance: 0
  Key-id   : 100
      AuthAlgorithm: aes-128-cmac
      Key-String   : *******
      Send Lifetime:-
          Start    : 02-16-2022 04:05:00 End         : Infinite
          Active   : Yes                 TimeToExpire: Infinite
          Timezone : GMT+00
```

Displaying Information on Current MACsec Sessions

You can display MACsec session activity for an interface, including the pre-shared key name, the most recent SAI information, and a list of peers.

1. For a quick overview of current MACsec sessions, enter the **show dot1x-mka sessions brief** command in privileged EXEC, global configuration, or dot1x-mka interface mode.

```
device(config)# show dot1x-mka sessions brief

Port    Link-Status  MKA-Status  Key-Server  Negotiated Capability

1/3/2   Down         Pending   ---      ---
1/3/3   Up           Secured   No       Integrity, Confidentiality with Off. 30
1/3/4   Up           Secured   No       Integrity, Confidentiality with Off. 30
```

2. To display full details on current MACsec sessions, in privileged EXEC, global configuration, or dot1x-mka interface mode, enter the **show dot1x-mka sessions ethernet** command followed by the interface identifier.

```
device(config)# show dot1x-mka sessions ethernet 1/3/3

Interface                : 1/3/3

  MACsec Status          : Secured
  DOT1X-MKA Enabled      : Yes
  DOT1X-MKA Active       : Yes
  Key Server             : No

Configuration Status:
  Enabled                : Yes
  Capability             :  Integrity,  Confidentiality
  Desired                : Yes
  Protection             : Yes
  Frame Validation       : Disable
  Replay Protection      : Strict
  Replay Protection Size : 0
  Cipher Suite           : GCM-AES-128
  Key Server Priority    : 20

  Local SCI              : 748ef8344a510082
  Member Identifier      : 802ed0536fcafc43407ba222
  Message Number         : 8612

Secure Channel Information:
  Latest SAK Status      : Rx & Tx
  Latest SAK AN          : 0
  Latest KI              : d08483062aa9457e7c2470e300000001
  Negotiated Capability  : Integrity, Confidentiality with offset 30

Peer Information:
State        Member Identifier           Message Number    SCI                 Priority
-----        -----------------           --------------    ----------------    --------
Live         d08483062aa9457e7c2470e3              8527    748ef83443910082          20
```

Displaying MKA Protocol Statistics for an Interface

You can display a report on MKA protocol activity for a particular interface.

In privileged EXEC, global configuration, or dot1x-mka interface mode, enter the **show dot1x-mka statistics ethernet** command to display MKA protocol statistics for the designated interface.

```
device(config-dot1x-mka-1/3/3)# show dot1x-mka statistics ethernet 1/3/3

Interface                : 1/3/3

MKA in Pkts              : 8585
MKA in SAK Pkts          : 1
MKA in Bad Pkts          : 0
MKA in Bad ICV Pkts      : 0
MKA in Mismatch Pkts     : 0
MKA out Pkts             : 8687
MKA out SAK Pkts         : 0
Number of SAK            : 1
```

Displaying MACsec Secure Channel Activity for an Interface

You can display currently enforced MACsec capabilities for a specific interface, along with secure channel statistics.

1. In privileged EXEC mode, enter the **clear macsec statistics** command for the designated interface.

   Results of the previous **show macsec statistics** command are removed.

2. In privileged EXEC, global configuration, or dot1x-mka interface mode, enter the **show macsec statistics** command to display information on MACsec configuration and secure channel activity for a particular interface.

   The following **show macsec statistics** command output is for an ICX 7450 device.

```
device# clear macsec statistics ethernet 10/2/1
device# show macsec statistics ethernet 10/2/1

Interface Statistics:
---------------------
rx Untag Pkts             : 1                   tx Untag Pkts             : 0
rx Notag Pkts             : 0                   tx TooLong Pkts           : 0
rx Badtag Pkts            : 0
rx Unknownsci Pkts        : 0
rx Nosci Pkts             : 0
rx Overrun Pkts           : 0

Transmit Secure Channels:
-------------------------

SA[0] Statistics:
Protected Pkts            : 0
Encrypted Pkts            : 4485

SA[1] Statistics:
Protected Pkts            : 0
Encrypted Pkts            : 0

SA[2] Statistics:
Protected Pkts            : 0
Encrypted Pkts            : 0

SA[3] Statistics:
Protected Pkts            : 0
Encrypted Pkts            : 0

SC Statistics:
Protected Octets          : 0                   Encrypted Octets          : 250473
Protected Pkts            : 0                   Encrypted Pkts            : 4485

Receive Secure Channels:
------------------------

SA[0] Statistics:
Ok Pkts                   : 3094                Invalid Pkts              : 0
Not using SA Pkts         : 0                   Unused Pkts               : 0
Not Valid Pkts            : 0

SA[1] Statistics:
Ok Pkts                   : 0                   Invalid Pkts              : 0
Not using SA Pkts         : 0                   Unused Pkts               : 0
Not Valid Pkts            : 0

SA[2] Statistics:
Ok Pkts                   : 0                   Invalid Pkts              : 0
Not using SA Pkts         : 0                   Unused Pkts               : 0
Not Valid Pkts            : 0

SA[3] Statistics:
Ok Pkts                   : 0                   Invalid Pkts              : 0
Not using SA Pkts         : 0                   Unused Pkts               : 0
Not Valid Pkts            : 0

SC Statistics:
OkPkts                    : 3094                Invalid Pkts              : 0
Not using SA Pkts         : 0                   Unused Pkts               : 0
Not Valid Pkts            : 0                   Unchecked Pkts            : 0
Delayed Pkts              : 0                   Late Pkts                 : 0
Valid Octets              : 0                   Decrypted Octets          : 157120
```

Configuring Logging and RADIUS Server Hosts

Logging Servers

When a logging server is configured, the syslogs are simultaneously stored locally and forwarded to the external syslog server if it is reachable.

If connectivity to the logging server is lost, the device automatically tries to reconnect.

Configuring an SSL Profile for Use with RADIUS Server Hosts

Configure an SSL profile for use with logging and RADIUS Server hosts for NDcPP.

You must configure an SSL profile, to be applied to the RADIUS server, for use in establishing a secure TLS connection. The SSL profile specifies the root (CA) certificate trustpoint and the remote domain name to be used in certification.

NOTE
On ICX 7250 and ICX 7450 series switches, the device trustpoint will not work with RADIUS due to the strict validation of certification. For TLS with RADIUS on these switches, an external certificate must be copied to the ICX device in order to establish a successful TLS session.

1. Name the SSL profile and enter profile configuration mode.

```
device# configure terminal
device(config)# ip ssl profile tls01
```

Syntax: ip ssl profile *profile-name*

Syntax: no ip ssl profile *profile-name*

2. Specify the trustpoint (CA server) that will be associated with the profile.

```
device(config-ssl-tls01)# trustpoint TLS-ABCD
```

Syntax: trustpoint *trustpoint-name*

Syntax: no trustpoint *trustpoint-name*

3. Configure the remote domain name that the FQDN of the remote network peer certificate issues to the server. This is the 'reference identifier' that must appear in the network peer's certificate.

   NOTE
   The ICX device expects the 'reference identifier' value to be either in CN, or, if SAN is present, this value must be shown as a DNS name in the SAN.

   NOTE
   The remote domain name must match the CN or SAN. ICX devices do not support wild card bits in SAN extensions.

```
device(config-ssl-tls01)# remotedomain ruckus.com
device(config-ssl-tls01)# exit
device(config)#
```

   Syntax: remotedomain *domain-name*

   Syntax: no remotedomain *domain-name*

4. (Optional) Use the **show ip ssl profile** command to check the user SSL profile and device SSL profile information.

The following example configures the SSL profile tls01 and associates it with the trustpoint TLS-ABCD with ruckus.com as the remote domain name that the end user certificate issues to the server.

```
device# configure terminal
device(config)# ip ssl profile tls01
device(config-ssl-tls01)# trustpoint TLS-ABCD
device(config-ssl-tls01)# remotedomain ruckus.com
```

Logging and RADIUS Server Host Configuration for NDcPP

A FastIron ICX device acting as a syslog client sends out audit logs over a trusted and secure tunnel.

To set up the secure tunnel and authenticate the syslog client and server, you must perform the following set of tasks:

- Configure PKI (Refer to Configuring PKI on page 50 in this guide for more information.)
- Configure one or both of the following:
  - Logging host
  - RADIUS server host for user authentication.

Configuring a Logging Host for NDcPP

Configure a logging host with the local or remote IP address of the syslog server and its remote port number. On the same line, specify a pre-configured SSL profile.

```
device# configure terminal
device(config)# logging host < ip-address | server-name > ssl-port < port-number > profile < profile-name >
```

Syntax: logging host { *ip-address* | *server-name* } **ssl-port** *port-number* **profile** *profile-name*

When audit logs are generated, the FastIron device establishes a secure TLS tunnel.

During the handshake with the server, the FastIron device receives the server certificate and obtains validation for the certificate from the CA server through the PKI infrastructure.

If validation is successful, the handshake continues to look for the client certificate. If the server has requested a client certificate, the FastIron device sends the client certificate, and the server validates it using Verify protocol logic.

If the client and server certificate validations are successful, the TLS tunnel is established, and audit logs are sent to the server over the secure and trusted tunnel. Subsequent log messages use the established TLS tunnel.

The following example configures a logging host with an IP address of 192.168.10.10. Its SSL port is 5002, and it uses the profile tls03.

```
device# configure terminal
device(config)# logging host 192.168.10.10 ssl-port 5002 profile tls03
```

Configuring a RADIUS Server Host for NDcPP

Configure a radius-server host for user authentication by specifying its local or remote IP address, the remote port of the RADIUS server, and the name of the previously configured SSL profile to be used.

```
device(config)# radius-server host < ip-address | server-name > ssl-auth-port < port-number > profile <
profile-name > authentication key < radius-key >
```

The following syntax statement includes only information relevant to encryption.

Syntax: radius-server { **host** { *ip-address* | *server-name* } **ssl-auth-port** *port-number* **profile** *profile-name* **authentication key** *radius-key* }

When the user tries to log into the FastIron device, he is first authenticated by the radius server. Before the FastIron device sends out the Radius request to the server, the FastIron device establishes a secure TLS tunnel.

During the handshake with the server, the FastIron device receives the server certificate, and it is validated by the CA server through the PKI infrastructure.

If validation is successful, the handshake continues to look for the client certificate. If the server has requested the client certificate, the FastIron device sends the client certificate, and the server validates it using Verify protocol logic.

If the client and server certificate validations are successful, a TLS tunnel is established, and the Radius authentication request and response are sent over the secure and trusted tunnel.

If TLS tunnel establishment fails, the FastIron device attempts to establish the tunnel and authenticate the user when the user tries to log in again.

The following example configures a RADIUS server host with an IP address of 10.20.158.104. The SSL authentication port is 8001. The profile used is tls03, and the RADIUS authentication key is tesT123$$.

```
device(config)# radius-server host 10.20.158.104 ssl-auth-port 8001 profile tls03 authentication key
tesT123$$
```

Syslog Messages

Syslog Messages in FIPS and CC Modes

The following table lists some of the syslog messages in FIPS and CC mode.

TABLE 9 Syslog Messages in FIPS and CC Modes

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FAU_GEN.1	Start-up and shut-down of the audit functions		Shut-down: 2022-04-12T20:02:49Z ICX7450-48P Router - - [meta sequenceId=10] BOM CLI CMD: "reload" by super user from console Start-up: 2022-04-12T20:00:34Z ICX7450-48P Router ICX7450_Router - System [meta sequenceId=9] BOM System: Interface ethernet mgmt1, state up
MACsecEP12:FCS_MACSEC_EXT.1	Session establishment	Secure Channel Identifier (SCI)	2022-04-28T17:26:59Z ICX7550-48P Router ICX7550_Router - General [meta sequenceId=58] BOM MACsec: communication is now secured for port 1/3/3 Session SCI - 3420e30008290082, CKN - acbf98acd980bf385193871209184001
MACsecEP12:FCS_MACSEC_EXT.3	e1: Creation and update of Secure Association Key	e1: Creation and update times	2022-04-28T17:26:59Z ICX7550-48P Router ICX7550_Router - General [meta sequenceId=60] BOM MACsec: new SAK generated for port 1/3/3
MACsecEP12:FCS_MACSEC_EXT.4	e4: Creation of Connectivity Association (per TD0509)	e4: Connectivity Association Key Names (per TD0509)	2022-04-28T17:26:59Z ICX7550-48P Router ICX7550_Router - General [meta sequenceId=58] BOM MACsec: communication is now secured for port 1/3/3 Session SCI - 3420e30008290082, CKN - acbf98acd980bf385193871209184001
NDcPP22e:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server	Added: 2022-04-29T19:24:59Z ICX7450-48P Router - - [meta sequenceId=17] BOM CLI CMD: "server 192.168.144.254 key ..... 4 " by super user from console Removed: 2022-04-29T19:24:53Z ICX7450-48P Router - - [meta sequenceId=14] BOM CLI CMD: "no server 192.168.144.254 key ..... 4 " by super user from console

TABLE 9 Syslog Messages in FIPS and CC Modes (continued)

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.	2022-03-25T16:44:51Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=4] BOM sshd: SSH access by user admin from src 192.168.144.254 rejected, 0 attempt(s) 2022-04-20T17:00:19Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=15] BOM sshd: Unable to negotiate with 192.168.144.254 port 41458: no matching cipher found. Their offer: aes128-gcm@openssh.com 2022-04-20T17:42:16Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=5] BOM sshd: Failed publickey for admin from 192.168.144.254 port 42462 ssh2
NDcPP22e:FCS_TLSC_EXT.1	Failure to establish a TLS Session.	Reason for failure.	2022-04-20T17:24:02Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=87] BOM System: SSL server connection failed for host 192.168.144.254 (null) 2022-04-05T20:53:46Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=7] BOM PKI: Trustpoint - rootca-rsa : Remote Domain name (bar.foo.example.com) validation failed with return code : 1 Cert ID - Serial No : 296, Issuer CN : subsubca-rsa, Subject CN : *.example.com 2022-04-18T18:11:13Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=79] BOM PKI: Trustpoint - rootca-rsa : Extended key usage validation failed - Unsupported certificate purpose Cert ID - Serial No : 161, Issuer CN : subsubca-rsa, Subject CN : tl28-16x.example.com 2022-04-19T21:34:12Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=17] BOM System: SSL connection failed to host 192.168.144.254, error: 14092105:SSL routines:SSL3_GET_SERVER_HELLO:wrong cipher returned 2022-04-20T17:27:27Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=193] BOM System: SSL server connection failed for host 192.168.144.254 6514 : "unknown cipher returned" 2022-04-22T21:46:07Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=264] BOM System: SSL server connection failed for host 192.168.144.254 6514 : "wrong ssl version" 2022-04-19T21:49:33Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=31] BOM System: SSL connection failed to host 192.168.144.254, error: 1408206D:SSL routines:SSL3_CHECK_CERT_AND_ALGORITHM:bad dh pub key length 2022-04-20T17:21:02Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=76] BOM System: SSL server connection failed for host 192.168.144.254 bad signature

TABLE 9 Syslog Messages in FIPS and CC Modes (continued)

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).	2022-04-13T15:17:23Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=16] BOM sshd: username TestUser is disabled
MACsecEP12:FIA_AFL.1	Administrator lockout due to excessive authentication failures		See NDcPP22e:FIA_AFL.1 above
NDcPP22e:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).	SSH Success: 2022-04-11T13:23:31Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=51] BOM Security: ssh login by super from src IP 192.168.144.254 from src MAC 0015.5d90.1701 to USER EXEC mode SSH Password Failure: 2022-03-25T16:44:51Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=4] BOM sshd: SSH access by user admin from src 192.168.144.254 rejected, 0 attempt(s) SSH Pubkey Failure: 2022-04-20T17:42:16Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=5] BOM sshd: Failed publickey for admin from 192.168.144.254 port 42462 ssh2 Console Success: 2022-04-11T13:22:16Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=17] BOM Security: console login by super to PRIVILEGED EXEC mode Console Failure: 2022-03-23T09:39:12Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=7] BOM login: TELNET access by user rejected for incorrect login
NDcPP22e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).	See NDcPP22e:FIA_UAU_EXT.2 above

TABLE 9 Syslog Messages in FIPS and CC Modes (continued)

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store	Cert Validation Failure: 2022-04-06T15:07:19Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=5] BOM PKI: Trustpoint - rootca-rsa : Certificate validation failed - self signed certificate in certificate chain Cert ID - Serial No : 157, Issuer CN : rootca-unacceptable-rsa, Subject CN : tl28-16x.example.com 2022-04-06T15:33:27Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=53] BOM PKI: Trustpoint - rootca-rsa : Certificate validation failed - certificate has expired Cert ID - Serial No : 19, Issuer CN : rootca-rsa, Subject CN : subca-expired-rsa 2022-04-06T15:38:10Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=71] BOM PKI: Trustpoint - rootca-rsa : OCSP - Certificate not valid, Reason : Revoked, code : -1 Cert ID - Serial No : 209, Issuer CN : subsubca-rsa, Subject CN : tl28-16x.example.com 2022-04-06T15:48:47Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=97] BOM PKI: Trustpoint - rootca-rsa : OCSP - OCSP purpose missing in responder certificate Cert ID - Serial No : 11, Issuer CN : rootca-rsa, Subject CN : subca-rsa 2022-04-06T16:02:32Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=152] BOM PKI: Trustpoint - rootca-rsa : Certificate validation failed - certificate signature failure Cert ID - Serial No : 103, Issuer CN : subsubca-rsa, Subject CN : tl28-16x.example.com 2022-04-20T17:16:32Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=70] BOM System: SSL server connection failed for host 192.168.144.254 wrong tag 2022-04-20T17:19:32Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=74] BOM System: SSL server connection failed for host 192.168.144.254 block type is not 01 Added Trust: 2022-02-16T16:31:13Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=123] BOM PKI: Trustpoint rootca-rsa: Successfully loaded certificate file rootca-rsa.pem Removed Trust: 2022-04-29T17:45:27Z ICX7450-48P Router ICX7450_Router - -[meta sequenceId=8] BOM PKI: Successfully deleted trustpoint rootca-rsa certificate file. Cert ID - Serial No : 65537, Issuer CN : rootca-rsa, Subject CN : rootca-rsa

TABLE 9 Syslog Messages in FIPS and CC Modes (continued)

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.		2022-04-06T19:52:50Z ICX7450-48P Router - - [meta sequenceId=184] BOM CLI CMD: "copy scp flash 192.168.144.254 /tmp/SPR09010bdevufi.bin primary" by super user from console 2022-04-06T19:52:57Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=185] BOM Download: COPY IMAGE TO FLASH START 2022-04-06T19:54:20Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=186] BOM Download: FIPS IMAGE VERIFICATION PASSED AND STORED IN FLASH 2022-04-06T19:54:20Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=187] BOM Download: COPY APPLICATION IMAGE FROM BUNDLE START 2022-04-06T19:54:20Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=188] BOM Download: COPY BOOTROM IMAGE FROM BUNDLE START 2022-04-06T19:54:41Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=189] BOM Download: COPY BUNDLE IMAGE COMPLETED
NDcPP22e:FMT_SMF.1	All management activities of TSF data.		Every admin command is logged and tagged with CLI_CMD so all maanagment functions are covered by that. Below is an example audit: 2022-04-11T12:59:50Z ICX7450-48P Router - - [meta sequenceId=6221] BOM CLI CMD: "logging buffered 50" by super user from conso
MACsecEP12:FMT_SMF.1	All management activities of TSF data.		See NDcPP22e:FMT_SMF.1 above
MACsecEP12:FPT_RPL.1	Detected replay attempt		2022-03-09T22:48:02Z ICX7550-48P Router MACSEC: Warning! Late-Pkts[Count 5] are received Port[1/3/1]
NDcPP22e:FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).	NTP: 2022-03-09T20:27:33Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=177] BOM Security: Time is updated by NTP server "192.168.144.254" from "20:27:32.524 GMT+00 Wed Mar 09 2022 " to "20:27:33.013 GMT+00 Wed Mar 09 2022 Manual: 2022-04-12T19:26:00Z ICX7450-48P Router ICX7450_Router - System [meta sequenceId=2782] BOM Clock Changed from old time 14:47:39.387 Eastern Tue Apr 12 2022 to new time 19:26:00.001 Eastern Tue Apr 12 2022 by super user from console session
NDcPP22e:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).		Success shown in FMT_MOF.1/ManualUpdate Failure: 2022-04-29T17:12:17Z ICX7450-48P Router ICX7450_Router - System [meta sequenceId=381] BOM Download: COPY BUNDLE IMAGE VALIDATION FAILED

TABLE 9 Syslog Messages in FIPS and CC Modes (continued)

Requirement	Audit Event	Audit Content	Sample Audit
NDcPP22e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.		2022-02-14T13:54:05Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=1] BOM Security: ssh timed out by admin from src IP 192.168.144.254 from src MAC 0015.5d90.1701 from USER EXEC mode
NDcPP22e:FTA_SSL.4	The termination of an interactive session.		SSH: 2022-04-11T12:12:09Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=5845] BOM Security: ssh logout by super from src IP 192.168.144.254 from src MAC 0015.5d90.1701 from PRIVILEGED EXEC mode Console: 2022-04-11T12:59:00Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=6217] BOM Security: console logout by super from PRIVILEGED EXEC mode
NDcPP22e:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.		2022-02-15T12:38:06Z ICX7450-48P Router ICX7450_Router - General [meta sequenceId=222] BOM Security: console timed out by admin from USER EXEC mode
NDcPP22e: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.	Initiation: 2022-04-06T20:04:44Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=32] BOM System: SSL server 192.168.144.254 6514 is now connected Termination: 2022-04-06T20:06:39Z ICX7450-48P Router ICX7450_Router - - [meta sequenceId=33] BOM System: SSL server 192.168.144.254 6514 is disconnected For failure see NDcPP22E:FCS_TLSC_EXT.1
NDcPP22e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.		See NDcPP22e:FIA_UAU_EXT.2 for SSH initiation See NDcPP22e:FTA_SSL.4 for SSH termination See NDcPP22e:FCS_SSHS_EXT.1 for SSH failure

OpenSSL License

OpenSSL License Overview

NOTE

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

NOTE

OpenSSL has been compiled without the Heartbeat extension.

License

This is a copy of the current LICENSE file inside the CVS repository.

```
LICENSE ISSUES
==============
    The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
    the OpenSSL License and the original SSLeay license apply to the toolkit.
    See below for the actual license texts. Actually both licenses are BSD-style
    Open Source licenses. In case of any license issues related to OpenSSL
    please contact openssl-core@openssl.org.

OpenSSL License
---------------

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1.Redistributions of source code must retain the above copyright
 *         notice, this list of conditions and the following disclaimer.
 *
 * 2.Redistributions in binary form must reproduce the above copyright
 *         notice, this list of conditions and the following disclaimer in
 *         the documentation and/or other materials provided with the
 *         distribution.
 *
 * 3.All advertising materials mentioning features or use of this
 *         software must display the following acknowledgment:
 *         "This product includes software developed by the OpenSSL Project
 *         for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4.The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *         endorse or promote products derived from this software without
 *         prior written permission. For written permission, please contact
 *         openssl-core@openssl.org.
 *
 * 5.Products derived from this software may not be called "OpenSSL"
 *         nor may "OpenSSL" appear in their names without prior written
 *         permission of the OpenSSL Project.
 *
 * 6.Redistributions of any form whatsoever must retain the following acknowledgment:
 *         "This product includes software developed by the OpenSSL Project
 *         for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
```

```
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1.Redistributions of source code must retain the copyright
 *         notice, this list of conditions and the following disclaimer.
 * 2.Redistributions in binary form must reproduce the above copyright
 *         notice, this list of conditions and the following disclaimer in the
 *         documentation and/or other materials provided with the distribution.
 * 3.All advertising materials mentioning features or use of this software
 *         must display the following acknowledgement:
 *         "This product includes cryptographic software written by
 *         Eric Young (eay@cryptsoft.com)"
 *         The word 'cryptographic' can be left out if the rouines from the library
 *         being used are not cryptographic related :-).
 * 4.If you include any Windows specific code (or a derivative thereof) from
 *         the apps directory (application code) you must include an acknowledgement:
 *         "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
```

```
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

350 West Java Dr., Sunnyvale, CA 94089 USA
https://www.commscope.com