Junos® OS

Common Criteria Configuration Guide for MX304 Device with JNP304-LMIC16 Line Card

Published
2025-05-29

RELEASE
22.4R2

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

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS Common Criteria Configuration Guide for MX304 Device with JNP304-LMIC16 Line Card
22.4R2

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

About This Guide

Use this guide to configure and evaluate the MX304 device with Junos OS Release 22.4R2 for Common Criteria (CC) compliance. Common Criteria for information technology are an international agreement signed by several countries that allows the evaluation of security products against a common set of standards. This standard ensures that the device meets global security requirements, enhancing its reliability and trustworthiness for users.

RELATED DOCUMENTATION

Common Criteria and FIPS Certifications

Compliance Advisor

1

CHAPTER

Overview

IN THIS CHAPTER

Common Criteria Overview

IN THIS SECTION

Common Criteria (CC) for information technology is an international agreement. Several countries have signed this agreement to assess the information technology products based on standardized criteria. In the Common Criteria Recognition Arrangement (CCRA), the participants have mutually agreed to recognize the product evaluations conducted in authorized laboratories in other countries. The authorized laboratories use a common methodology for information technology security evaluation.

CC Configuration Overview

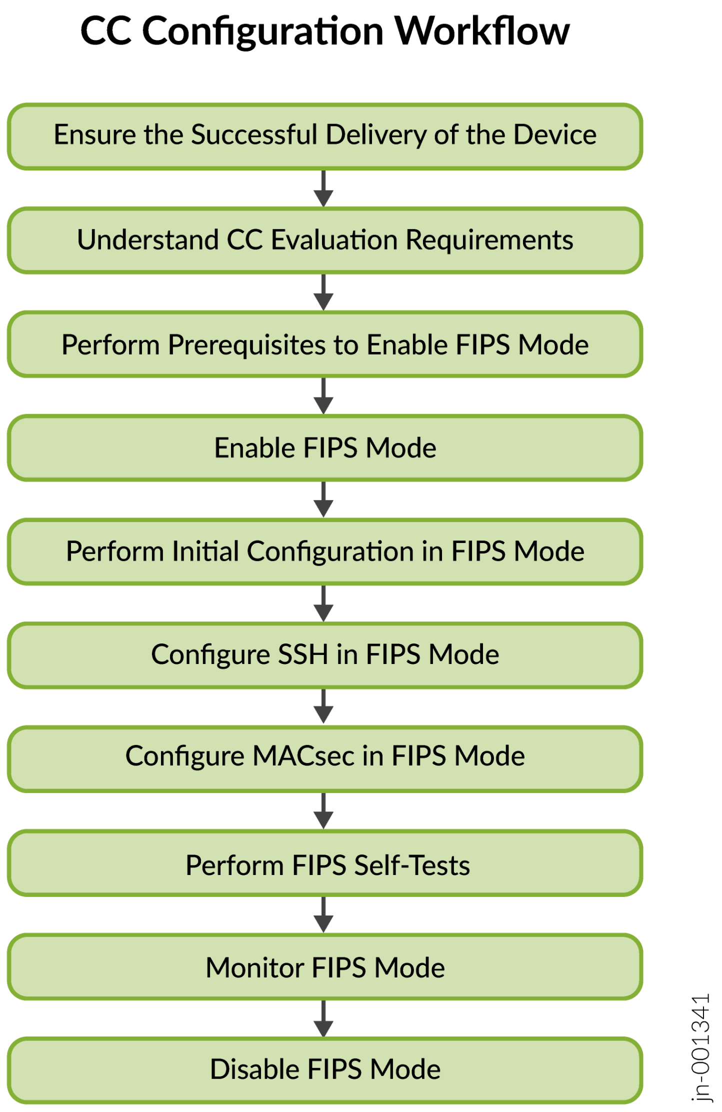
CC configuration provides the steps required to duplicate the CC compliance evaluation configuration for your device with Junos OS. This configuration is referred to as the evaluated configuration. This evaluated configuration ensures your device, sometimes referred as target of evaluation (TOE), adheres to the evaluation standards mentioned in Table 1 on page 2.

Table 1: CC Evaluation Standards

Evaluation Standard	Description
Network Device Collaborative Protection Profile (NDcPP) v2.2e—https://www.niap-ccevs.org/MMO/PP/CPP_ND_V2.2E.pdf	Provides the CC collaborative Protection Profile (cPP) to express the security functional requirements and security assurance requirements for a network device.
Media Access Control Security (MACsec) Extended Package (EP) 1.0—https://www.niap-ccevs.org/MMO/PP/pp_ndcpp_macsec_ep_v1.0.pdf	Describes the security requirements for a network device that implement MACsec encryption to secure communications over a trusted channel.

Figure 1 on page 3 shows the workflow of CC configuration on your device:

Figure 1: CC Configuration Workflow


CC Configuration Workflow
Ensure the Successful Delivery of the Device
Understand CC Evaluation Requirements
Perform Prerequisites to Enable FIPS Mode
Enable FIPS Mode
Perform Initial Configuration in FIPS Mode
Configure SSH in FIPS Mode
Configure MACsec in FIPS Mode
Perform FIPS Self-Tests
Monitor FIPS Mode
Disable FIPS Mode
jn-001341

Supported Platform

This document evaluates the following platform for CC compliance:

- MX304 with JNP304-LMIC16 line card (https://www.juniper.net/us/en/products/routers/mx-series/mx304-universal-routing-platform.html)

For the details of Juniper Networks' certified products, see Compliance Advisor.

2
CHAPTER

Prerequisites to Enable FIPS Mode

IN THIS CHAPTER

Ensure Successful Delivery of the Device

When you receive the device, you must verify that it has not been tampered with during the delivery process. Perform the tasks mentioned in Table 2 on page 6when you receive the device to confirm its integrity.

Table 2: Successful Delivery of the Device

Checks to Perform	Details
Purchase Order	Order the device using a purchase order. Juniper Networks always requires a purchase order before shipping its devices.
Shipment Notification	Ensure that you receive the shipment notification that Juniper Networks sends to your provided e-mail address. Confirm that the e-mail includes: • Purchase order number • Juniper Networks order number used to track the shipment • Carrier tracking number used to track the shipment • List of items shipped, including serial numbers of each item • Address and other contact information of both the supplier and the customer
Shipment Initiation	To confirm the shipment initiation from Juniper Networks: • Compare the carrier tracking number in the Juniper Networks shipping notification with the tracking number on the package received. • Log in to the Juniper Networks online customer support portal at https://support.juniper.net/support/ to view the order status.
Shipping Label	Ensure that the shipping label correctly identifies the customer name, customer address, and the device.

Table 2: Successful Delivery of the Device *(Continued)*

Checks to Perform	Details
Outside Packaging	• Examine the outer shipping box and tape. • Ensure that the shipping tape is intact and has not been tampered with or cut. • Confirm that the box has not been cut or damaged in a way that would allow access to the device.
Inside Packaging	• Examine the plastic bag and its seal to ensure that the bag has not been cut or removed. • Verify that the seal is intact.

If you identify any issues during inspection, immediately contact the supplier. Provide the order number, carrier tracking number, and a description of the identified problem to the supplier.

Management Interfaces Overview

You can use management interfaces on Juniper Networks devices to configure, monitor, and troubleshoot the devices. These interfaces are critical for maintaining the health and performance of the network. Table 3 on page 7 describes the management interfaces that you can use to evaluate the device configurations for CC compliance.

Table 3: Management Interfaces

Interface	Details
Local management interfaces	The RJ-45 console port on the device functions as RS-232 DTE. You can use the CLI over this port to configure the device from a terminal.
Remote management protocols	You can manage the device remotely over any Ethernet interface. SSHv2 is the only remote management protocol that you can use in the Common Criteria (CC) evaluated configuration. The remote management protocols J-Web and Telnet are not available to use on the device.

Download Software Packages from Juniper Networks

You can download the Junos OS software package from the Juniper Networks website. Before you begin to download the software, ensure that you have a Juniper Networks user account and a valid support contract. To create an account, complete the registration form at the Juniper Networks website: https://userregistration.juniper.net/.

To download software packages from Juniper Networks:

1. Using a browser, navigate to https://support.juniper.net/support/downloads/.
2. Log in with your username (generally your e-mail address) and password supplied by Juniper Networks representatives.
3. Download the software package you need. See Downloading Software (Junos OS).

RELATED DOCUMENTATION

Installation and Upgrade Guide

Install Junos OS Software Package

To install the Junos OS software package on your device:

1. Download the software package as described in "Download Software Packages from Juniper Networks" on page 8.
2. Connect the console port on your Juniper Networks device to your management device and log in to the Junos OS CLI.
3. (Optional) Back up the current software configuration to a second storage option. See the *Software Installation and Upgrade Guide* for instructions on performing this task.
4. (Optional) Copy the downloaded software package to your device. We recommend to use FTP to copy the file to the **/var/tmp/** directory.

   This step is optional because you can also install Junos OS when the software image is available at a remote location. These instructions describe the software installation process for both scenarios.

5. Use the following command to install the new package on the device:

   ```
   user@host> request vmhost software add package
   ```

   Replace the variable option ***package*** with one of the following paths:

   - For a software package in a local directory on the device, use **/var/tmp/*package*.tgz**.

   - For a software package on a remote server, use one of the following paths, replacing the variable option *package* with the software package name.

     - **ftp://*hostname*/*pathname*/*package*.tgz**

     - **https://*hostname*/*pathname*/*package*.tgz**

   Junos OS comes in signed packages with digital signatures, ensuring that Juniper Networks software operates flawlessly. During installation of the software packages, Junos OS validates the signatures and the public key certificates used to sign the software packages digitally. If the signature or certificate is found invalid (for example, an expired certificate or the system fails to verify the certificate against the root certificate authority (CA) certificate stored in the Junos OS internal store), the installation process fails.

6. Reboot the device to load the installation:

   ```
   user@host> request vmhost reboot
   ```

7. After the system reboots, log in to the device. Use the `show version` command to verify the successful installation of the desired software version.

   ```
   user@host> show version
   Hostname: hostname
   Model: mx304
   Junos: 22.4R2.8
   JUNOS OS Kernel 64-bit [20230321.be5f9c0_builder_stable_12_224]
   JUNOS OS libs [20230321.be5f9c0_builder_stable_12_224]
   JUNOS OS runtime [20230321.be5f9c0_builder_stable_12_224]
   JUNOS OS time zone information [20230321.be5f9c0_builder_stable_12_224]..........[Continue]
   ```

Zeroize the System to Clear System Data for FIPS Mode

IN THIS SECTION

Zeroization is a security process used in computing and cryptography to securely erase sensitive data from memory, storage devices, or cryptographic modules. The goal is to prevent unauthorized access to this data ensuring that not even the most advanced forensic techniques cannot recover or reconstruct the data.

In FIPS mode, the Security Administrator initiates zeroization. Zeroization completely erases all configuration information about the Routing Engines, including all plain-text passwords, secrets, private keys for SSH, local encryption details, and local authentication details.

Note that, in reference to cryptographic key destruction, TOE does not support delayed key destruction.

⚠ **CAUTION**: Perform system zeroization with care. Zeroization eliminates all data from the Routing Engine and the device returns to the factory-default state, without any configured users or configuration files.

Why Zeroize Your Device for FIPS Mode?

For FIPS 140-3 compliance, you must zeroize the system to remove sensitive information before enabling or disabling FIPS mode on the device.

You must enter or reenter all the critical security parameters (CSPs) on your devices in FIPS mode to consider it as a valid FIPS cryptographic module.

When to Zeroize Your Device for FIPS Mode?

You can zeroize the device:

- **Before enabling FIPS mode of operation**: Perform zeroization before enabling FIPS mode to prepare your device to operate as a FIPS cryptographic module.

- **Before disabling FIPS mode of operation**: Perform zeroization before disabling FIPS mode on the device to begin repurposing your device for non-FIPS operation.

> **NOTE**: Juniper Networks does not support non-FIPS software installation in a FIPS environment, but doing so might be necessary in certain test environments. Ensure that you zeroize the device before proceeding.

Zeroize the System for FIPS Mode

Follow this procedure to zeroize your device:

1. Log in to the device as Security Administrator from the CLI, then enter:

```
security-administrator@host> request vmhost zeroize no-forwarding
VMHost Zeroization : Erase all data, including configuration and log files ? [yes,no] (no)
yes
```

2. To initiate the zeroization process, type **yes** at the prompts:

```
Erase all data, including configuration and log files?   [yes, no] (no) yes
VMHost Zeroization : Erase all data, including configuration and log files ? [yes,no] (no)
yes

warning: Vmhost will reboot and may not boot without configuration
warning: Proceeding with vmhost zeroize
Zeroise secondary internal disk ...
Proceeding with zeroize on secondary disk
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of secondary disk completed
Zeroize primary internal disk ...
Proceeding with zeroize on primary disk
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_rsa_key.pub
/etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_rsa_key
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of primary disk completed
Zeroize done
warning: Proceeding with vmhost reboot
Initiating vmhost reboot...
```

The entire operation can take considerable time depending on the size of the media, but the system removes all the CSPs within a few seconds. The physical environment must remain secure until the zeroization process is complete.

3

CHAPTER

Enable FIPS Mode

IN THIS CHAPTER

Junos OS in FIPS Mode Overview

IN THIS SECTION

FIPS 140-3 defines security levels for the hardware and software that performs cryptographic functions. Enable FIPS mode from the Junos OS CLI to operate your devices in a FIPS 140-3 Level 1 environment.

The Security Administrator enables FIPS mode in Junos OS and sets up keys and passwords for the system and other FIPS users.

Cryptographic Boundary on Your Device

The cryptographic boundary is a clearly defined physical or logical perimeter that encompasses all components that are necessary for the secure operation of the cryptographic module. The components within a cryptographic boundary include processors, memory, interfaces, and any other hardware or software that contributes to the module's cryptographic functions.

FIPS 140-3 compliance requires a defined cryptographic boundary around each cryptographic module on a device. Junos OS in FIPS mode prevents the cryptographic module from running any software that is not a part of the FIPS-certified distribution. It allows only FIPS-approved cryptographic algorithms. No CSPs, such as passwords and keys, can cross the cryptographic boundary of the module in unencrypted format.

⚠ **CAUTION**: FIPS mode does not support Virtual Chassis features. Avoid configuring Virtual Chassis in FIPS mode.

How FIPS Mode Differs from Non-FIPS Mode

Table 4 on page 15 summarizes how Junos OS in FIPS mode differs from Junos OS in non-FIPS mode:

Table 4: FIPS Mode and Non-FIPS Mode Comparison

Features	FIPS Mode	Non-FIPS Mode
Self-tests of all cryptographic algorithms at startup	Yes	No
Self-tests of random number and key generation perform continuously	Yes	No
Weak cryptographic algorithms such as Data Encryption Standard (DES) and MD5	Not allowed	Allowed
Weak, remote, or unencrypted management connections	Not allowed	Allowed
Local and unencrypted console access across all modes of operation	Allowed	Allowed
One-way algorithm used for password hashing	Yes	Yes
Administrator passwords with less than 10 characters length	Not allowed	Allowed
You must encrypt cryptographic keys before transmission	Yes	Not necessary

FIPS Terminology

Understand the FIPS-related terms and supported algorithms to perform tasks with Junos OS in FIPS mode.

Table 5: FIPS Terminology

Terminology	Description
CSP	Critical security parameter (CSP) is security-related information. For example, secret and private cryptographic keys and authentication data such as passwords and personal identification numbers (PINs). The disclosure or modification of this information can compromise the security of a cryptographic module or the information that the module protects. For details, see "Critical Security Parameters in FIPS Mode" on page 24.
Cryptographic module	The set of hardware, software, and firmware that implements approved security functions (including cryptographic algorithms and key generation) is contained within a defined boundary. Cryptographic module is validated to meet specific security requirements.
FIPS	Federal Information Processing Standard (FIPS) 140-3 specifies the requirements for security and cryptographic modules. Junos OS in FIPS mode complies with FIPS 140-3 Level 1.
Hashing	A message authentication method that applies a cryptographic technique iteratively to a message of an arbitrary length and produces a hash message digest or signature of a fixed length that is then appended to the sent message.
KAT	Known answer test (KAT) is the system self-tests that validate the output of cryptographic algorithms approved for FIPS and verify the integrity of some Junos OS modules. For details, see "Known Answer Test (KAT)" on page 78.
NDcPP	Collaborative Protection Profile for Network Devices (NDcPP) is a set of security requirements and guidelines designed to ensure the security and robustness of network devices. These devices include routers, switches, firewalls, and other hardware that manage network traffic.
Security Administrator	A user with appropriate permissions who is responsible for securely enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode of operation on a device. For details, see "FIPS Mode Roles and Services for Junos OS" on page 28.
SSH	SSH is a protocol that uses strong authentication and encryption for remote access across a non-secure network. SSH provides remote login, remote program execution, file copy, and other functions. The protocol is intended as a secure replacement for `rlogin`, `rsh`, and `rcp` in a UNIX environment. Use SSHv2 for CLI configuration to secure the information sent over administrative connections. By default, Junos OS enables SSHv2 and disables SSHv1 because of security concerns.

Table 5: FIPS Terminology *(Continued)*

Terminology	Description
Zeroization	Zeroizing a device erase all CSPs and other user-created data on the device before its operation as a FIPS cryptographic module or in preparation for repurposing the devices for non-FIPS operation. See "Zeroize the System to Clear System Data for FIPS Mode" on page 10.

Cryptographic Algorithms and Protocols Supported in FIPS Mode

Table 6 on page 17 lists the cryptographic algorithms and protocols supported on your device in FIPS mode.

Table 6: Cryptographic Algorithms and Protocols Supported in FIPS Mode

Protocol	Key Exchange	Authentication	Cipher	Integrity
SSHv2	• ECDH-sha2-nistp256 • ECDH-sha2-nistp384 • ECDH-sha2-nistp521	Host (module): • ECDSA P-256 • SSH-RSA Client (user): • ECDSA P-256 • ECDSA P-384 • ECDSA P-521 • SSH-RSA • RSA-SHA2-256 • RSA-SHA2-512	• AES CTR 128 • AES CTR 256 • AES CBC 128 • AES CBC 256	• HMAC-SHA-1 • HMAC-SHA-256 • HMAC-SHA-512

Table 7 on page 18 summarizes the details of supported cryptographic algorithms in FIPS mode. Symmetric methods use the same key for encryption and decryption, whereas asymmetric methods use different keys for encryption and decryption.

Table 7: Supported Cryptographic Algorithms

Cryptographic Algorithms	Description
AES	Advanced Encryption Standard (AES) algorithm is defined in FIPS PUB 197. The AES algorithm uses keys of 128 or 256 bits to encrypt and decrypt data in blocks of 128 bits.
ECDH	Elliptic Curve Diffie-Hellman (ECDH) is a variant of the Diffie-Hellman key exchange algorithm that uses cryptography based on the algebraic structure of elliptic curves over finite fields. ECDH enables two parties, both possessing an elliptic curve public-private keypair, to create a shared secret over an insecure channel. You can use the shared secret as a key or to derive another key for encrypting subsequent communications using a symmetric key cipher.
ECDSA	Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the Digital Signature Algorithm (DSA) that uses cryptography based on the algebraic structure of elliptic curves over finite fields. The bit size of the elliptic curve determines the difficulty of decrypting the key. The public key believed to be needed for ECDSA is about twice the size of the security level, in bits. You can configure ECDSA using the P-256, P-384, and P-521 curves under OpenSSH.
HMAC	Hash-based Message Authentication Code (HMAC) defined as “Keyed-Hashing for Message Authentication” in RFC 2104. HMAC combines hashing algorithms with cryptographic keys for message authentication. For Junos OS in FIPS mode, HMAC uses the iterated cryptographic hash functions SHA-1, SHA-256, and SHA-512 along with a secret key.
SHA-256 and SHA-512	SHA belonging to the SHA-2 standard defined in FIPS PUB 180-2. Developed by NIST, SHA-256 produces a 256-bit hash digest and SHA-512 produces a 512-bit hash digest.

Password Specifications and Guidelines for Junos OS in FIPS Mode

The Security Administrator needs to ensure that all user passwords conform to Junos OS in FIPS mode requirements in Table 8 on page 19. Attempts to configure passwords that do not conform to these specifications result in an error.

Table 8: Password Specifications and Guidelines

Requirements	Details
Length	Passwords must contain at least 10 characters.
Character set	Passwords must contain at least three of the following five character sets: • Uppercase letters • Lowercase letters • Digits • Punctuation marks • Keyboard characters are not included in the other four sets. These characters are the percent sign (%) and the ampersand (&). **NOTE**: Avoid control characters in passwords.
Authentication	All passwords and keys used to authenticate peers must contain at least 10 characters. In some cases, the number of characters must match the digest size.
Password encryption	To change the default encryption method (SHA512), include the `format` statement at the `[edit system login password]` hierarchy level.

Table 9 on page 20 summarizes the guidelines for strong passwords and the characteristics of weak passwords.

Table 9: Guidelines for Strong Passwords and Characteristics of Weak Passwords

Guidelines for Strong Passwords	Characteristics of Weak Passwords
You can create strong and reusable passwords by using letters from a favorite phrase or word and concatenating these letters with unrelated words, digits, and punctuation marks.	Weak passwords typically exhibit several key characteristics that make the password easy to guess or crack, thereby compromising the security of an account or system. Avoid using the weak passwords.
Strong passwords are made up of alphanumeric characters and punctuation. For FIPS compliance, include at least one change of case, one or more digits, and one or more punctuation marks in the password.	Words that might be found in or exist as a permuted form in a system file such as `/etc/passwd`.
Strong passwords are easy to remember so that you are not tempted to write it down.	The hostname of the system (always a first guess)
You must change the passwords periodically	Any word or phrase that appears in a dictionary or a well-known source, including dictionaries and thesaurus in languages other than English; works by classical or popular writers; or common words and phrases from sports, sayings, movies, or television shows. Permutations of any words or phrases mentioned above. For example, a dictionary word with letters replaced with digits (`r00t`) or with digits added at the end of the word.
You must not disclose the passwords to anyone	Any machine-generated password. Algorithms reduce the search space of password-guessing programs, and you must not use machine-generated password.

Configure FIPS Mode on Your Device

You must establish the passwords conforming to the FIPS password requirements. When you enable FIPS mode in Junos OS on your device, you must follow the password specification guidelines, see "Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 19.

To enable FIPS mode on your device:

1. Zeroize the device to delete all CSPs before enabling FIPS mode. See "Zeroize the System to Clear System Data for FIPS Mode" on page 10.
2. After the device comes up in `Amnesiac` mode, log in using the username `root` and password `""` (blank).

   ```
   FreeBSD/amd64 (Amnesiac) (ttyu0)
   login: root
   --- JUNOS 22.4R2.8 Kernel 64-bit  JNPR-12.1-20230321.be5f9c0_buil
   root@host:~ # cli
   root>
   ```

3. Configure root authentication with a password of at least 10 characters.

   ```
   root@host> edit
   Entering configuration mode
   [edit]
   root@host# set system root-authentication plain-text-password
   New password:
   Retype new password:
   [edit]
   root@host# commit
   commit complete
   ```

4. The `fips-mode` and `jpfe-fips` packages are optional for enabling FIPS. These packages are part of the Junos OS software. Use the following commands to enable the software packages:

   ```
   root@host> request system software add optional://fips-mode
   Verified fips-mode signed by PackageProductionECP256_2022 method ECDSA256+SHA256
   ```

5. Configure the chassis boundary.

   ```
   [edit]
   root@hostname# set system fips chassis level 1
   ```

6. Commit the configuration.

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

```
system reboot is required to transition to FIPS level 1
commit complete
```

7. Reboot the device.

   ```
   [edit]
   root@host# run request vmhost reboot
   ```

   After rebooting the device, the system performs the FIPS self-tests and enters the FIPS mode.

   ```
   root@host:fips>
   ```

Operational Environment for Junos OS in FIPS Mode

IN THIS SECTION

A Juniper Networks device with Junos OS in FIPS mode creates a hardware and software operational environment different from the environment of a device in non-FIPS mode.

Hardware Environment for Junos OS in FIPS Mode

Junos OS in FIPS mode establishes a cryptographic boundary in the device, preventing any CSP from crossing the boundary in plain text. Every hardware component in the device, requiring a cryptographic boundary for FIPS 140-3 compliance, acts as a separate cryptographic module. Junos OS in FIPS mode includes two types of hardware with cryptographic boundaries: one specifically for each Routing Engine and another for the entire chassis.

Cryptographic methods are not a substitute for physical security. You must place the hardware in a secure physical environment. All types of users have the responsibility to keep keys or passwords secret and prevent unauthorized personnel from viewing written records.

Software Environment for Junos OS in FIPS Mode

A Juniper Networks device with Junos OS in FIPS mode creates a special type of nonmodifiable operational environment. To achieve this environment on the device, the system prevents the execution of any binary file that was not part of the certified Junos OS in FIPS mode distribution. When a device is in FIPS mode, it can use only Junos OS.

The Security Administrator establishes the software environment for Junos OS in FIPS mode after successfully enabling FIPS mode on the device. The Juniper Networks website hosts the Junos OS image that includes FIPS mode, and you can install the image on a functioning device.

For FIPS 140-3 compliance, we recommend you delete all user-created files and data by zeroizing the device before enabling FIPS mode.

Enabling FIPS mode disables many of the usual Junos OS protocols and services. You cannot configure the following services in Junos OS in FIPS mode:

- finger
- ftp
- rlogin
- telnet
- tftp
- xnm-clear-text

Attempts to configure these services or load configurations with these services configured, result in a configuration syntax error. You can use only SSH as a remote access service.

All passwords established for users after the upgrade to Junos OS in FIPS mode must conform to the specifications for Junos OS in FIPS mode. For more information, see "Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 19.

Avoid attaching the device to a network until the Security Administrator completes configuring the device using the local console connection.

For strict compliance, while using Junos OS in FIPS mode, avoid examining core and crash file information about the local console, as the information can include some CSPs in plain text.

Critical Security Parameters in FIPS Mode

IN THIS SECTION

Critical Security Parameters

Critical security parameters (CSPs) are vital pieces of information or data for the security of cryptographic systems. CSPs include any information that requires protection to ensure the confidentiality, integrity, and authenticity of cryptographic operations.

Disclosing or modifying the CSPs can compromise the security of the cryptographic module and the information that protects.

Zeroization of your device erases all traces of CSPs in preparation for operating the device or Routing Engine as a cryptographic module.

Table 10 on page 24 lists and describes the CSPs on your device.

Table 10: Critical Security Parameters

CSP	Description	Zeroize	Use
SSHv2 private host key	An ECDSA or RSA key used to identify the host, generated the first time SSH is configured.	Zeroize command	Identifies the host

Table 10: Critical Security Parameters *(Continued)*

CSP	Description	Zeroize	Use
SSHv2 session keys	Session keys used with SSHv2 and as a Diffie-Hellman private keys. Encryption: AES-128, AES-256. MAC: HMAC-SHA-1, HMAC-SHA-2-256, HMAC-SHA2-512. Key exchange: ECDH-sha2-nistp256, ECDH-sha2-nistp384, and ECDH-sha2-nistp521.	Power cycle and session termination	Symmetric key used to encrypt data between host and client.
User authentication key	Hash of the user's password: SHA256 and SHA512	Zeroize command	Authenticates a user to the cryptographic module.
Security Administrator authentication key	Hash of the Security Administrator's password: SHA256, SHA512.	Zeroize command	Authenticates the Security Administrator to the cryptographic module.
HMAC DRBG seed	Seed for deterministic random bit generator (DRBG).	Seed is not stored by the cryptographic module.	A seed for DRBG
HMAC DRBG V value	The value (V) of the output block length (outlen) in bits, which is updated each time another outlen bits of output is produced.	Power cycle	A critical value of the internal state of DRBG.
HMAC DRBG key value	The current value of the outlen-bit keys, which is updated at least once each time that the DRBG mechanism generates pseudorandom bits.	Power cycle	A critical value of the internal state of DRBG.

Table 10: Critical Security Parameters *(Continued)*

CSP	Description	Zeroize	Use
NDRNG entropy	The NDRNG provides 448 bits of entropy collected per NIST SP 800-90 B. The NDRNG provides 448 bits of entropy collected per NIST SP 800-90 B from the Junos kernel software entropy source to seed the DRBG. The entropy is conditioned using a vetted conditioning component (SHA-512) and reseeds the DRBG whenever an additional 448 bits of entropy are collected.	Power cycle	A critical value of the internal state of DRBG.

In FIPS mode, all CSPs must enter and exit the cryptographic module in encrypted form. Any CSP encrypted with a non-approved algorithm is considered plain text by FIPS.

The system hashes the local passwords with the SHA256 or SHA512 algorithm. Password recovery is not possible in FIPS mode, and you cannot boot into single-user mode without the correct root password.

4

CHAPTER

Initial Configuration in FIPS Mode

IN THIS CHAPTER

FIPS Mode Roles and Services for Junos OS

SUMMARY

Learn about FIPS mode roles and services for Junos OS.

IN THIS SECTION

FIPS Mode Roles and Services

IN THIS SECTION

In FIPS mode, a role refers to the specific functions or responsibilities that users have when interacting with the cryptographic module. The primary roles in FIPS mode include:

1. Security Administrator
2. FIPS user

The Security Administrator and FIPS users perform all configuration tasks for Junos OS in FIPS mode and issue all statements and commands. Security Administrator and FIPS user configurations must meet the requirements for Junos OS in FIPS mode.

The Junos OS in non-FIPS mode provides a wide range of capabilities for users and supports identity-based authentication.

Security Administrator Role and Responsibilities

The Security Administrator role is associated with the defined login class `security-admin`. A Security Administrator has the necessary permissions to perform all tasks to manage Junos OS. The system requires administrative users (Security Administrator) to provide unique identification and authentication data before granting any administrative access.

BEST PRACTICE: We recommend that the Security Administrator follows security measures such as keeping passwords secure and checking audit files.

The permissions that distinguish the Security Administrator from other FIPS users are `secret`, `security`, `maintenance`, and `control`. The Security Administrator has the login class that contains all these permissions.

The Security Administrator role is crucial for maintaining the integrity and security of the system, especially in environments that require adherence to stringent federal security standards.

The Security Administrator has the following responsibilities:

1. Administer locally and remotely.
2. Create, modify, and delete user accounts, including configuration of authentication failure parameters.
3. Re-enable a user account.
4. Configure and maintain cryptographic elements related to the establishment of secure connections to and from the evaluated product.
5. Reset user passwords with FIPS-approved algorithms.
6. Examine log and audit files for events of interest.
7. Erase user-generated files, keys, and data by zeroizing the device.

FIPS User Role and Responsibilities

A FIPS user is defined as any user who does not have the `secret`, `security`, `maintenance`, and `control` permissions.

All FIPS users, including the Security Administrator, can view the configurations in the system. However, only the user assigned as the Security Administrator can modify the configurations. All FIPS users can view status output, but only the Security Administrator can reboot or zeroize the device.

What Is Expected of all FIPS Users

All FIPS users, including the Security Administrator, must always follow the security guidelines and ensure to:

- Keep all passwords confidential.
- Store devices and documentation in a secure area.
- Deploy devices in secure areas.
- Check audit files periodically.
- Conform to all other FIPS 140-3 security rules.
- Ensure device security always.

Configure Security Administrator Login Access

Junos OS in FIPS mode offers a finer granularity of user permissions than those mandated by FIPS 140-3. For FIPS 140-3 compliance, any FIPS user with the `secret`, `security`, `maintenance`, and `control` is a Security Administrator. In most cases, the `super-user` class suffices for the Security Administrator.

Junos OS login classes define the access privileges and permissions for using CLI commands and statements. For details, see Login Classes Overview.

To configure login access for a Security Administrator:

1. Log in to the device with the root password and enter configuration mode:

   ```
   root@host# edit
   Entering configuration mode
   [edit]
   root@host#
   ```

2. Name the user as `security-administrator`. Assign the Security Administrator:

   - A user ID that must be a unique number associated with the login account in the range of 100 through 64000
   - A class, for example, `super-user`

- When you assign the class, you assign the permissions such as `secret`, `security`, `maintenance`, and `control`.

```
[edit]
root@host# set system login user username uid value class class-name
```

For example:

```
[edit]
root@host# set system login user security-administrator uid 6400 class super-user
```

3. Assign the Security Administrator a plain-text password for login authentication, see "Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 19. Set the password by typing a password after the prompts `New password` and `Retype new password`.

```
[edit]
root@host# set system login user username class class-name authentication (plain-text-
password | encrypted-password)
```

For example:

```
[edit]
root@host# set system login user security-administrator class super-user authentication plain-
text-password
```

4. Optionally, display the configuration:

```
[edit]
root@host#edit system
[edit system]
root@host#show
login {
    user security-administrator {
        uid 6400;
        authentication {
            encrypted-password "<cipher-text>"; ## SECRET-DATA
        }
        class super-user;
    }
}
```

5. Commit the changes and exit the configuration mode:

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

Configure FIPS User Login Access

As a Security Administrator, you can create FIPS users. The system does not permit FIPS users to have the permissions that are given to the Security Administrator only—for example, the permission to zeroize the system.

To configure login access for a FIPS user:

1. Log in to the device with your Security Administrator password and enter configuration mode:

```
security-administrator@host:fips> edit
Entering configuration mode
[edit]
security-administrator@host:fips#
```

2. Assign the user:

   - A username

   - User ID, which must be a unique number in the range of 1 through 64000

   - A Class—When you assign the class, you assign the permissions such as `clear`, `network`, `resetview`, and `view-configuration`.

```
[edit]
security-administrator@host:fips# set system login user username uid value class class-name
```

For example:

```
[edit]
security-administrator@host:fips# set system login user fips-user1 uid 6401 class read-only
```

3. Following the guidelines in "Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 19, assign the FIPS user a plain-text password for login authentication. Set the password by typing a password after the prompts `New password` and `Retype new password`.

```
[edit]
security-administrator@host:fips# set system login user username class class-name
authentication (plain-text-password | encrypted-password)
```

For example:

```
[edit]
security-administrator@host:fips# set system login user fips-user1 class read-only
authentication plain-text-password
```

4. Optionally, display the configuration:

```
[edit]
security-administrator@host:fips# edit system
[edit system]
security-administrator@host:fips# show
login {
    user fips-user1 {
        uid 6401;
        authentication {
            encrypted-password "<cipher-text>"; ## SECRET-DATA
        }
        class read-only;
    }
}
```

5. Commit the changes and exit the configuration mode:

```
[edit]
security-administrator@host:fips# commit
security-administrator@host:fips#  exit
```

Configure Password Rules for an Authorized Administrator

An account for root is always present in the device and it is not intended for use in normal operation. The FIPS-evaluated configuration restricts the root account from performing initial installation and configuring the evaluated device.

The authorized administrator is associated with a defined login class and has all the permissions. The system stores the data locally for fixed password authentication.

Follow the guidelines in "Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 19 when you provide passwords for authorized administrator accounts. Define the password specifications rules for an authorized administrator:

- Define the minimum password length requirement as 10 characters.

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

- The password can have:

  - Both punctuation marks and other special characters
  - Combination of uppercase letter, lowercase letters, numbers, and special characters such as **!**, **@**, **#**, **$**, **%**, **^**, **&**, *****, **(**, and **)**

  The password must include at least one uppercase letter and one lowercase letter.

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

- Define the minimum number of character sets or character set changes. Each plain-text password requires at least three character sets.

```
[ edit ]
security-administrator@host:fips# set system login password minimum-changes 3
```

- The hashing algorithm for user passwords can be either SHA256 or SHA512. SHA512 is the default hashing algorithm.

```
[ edit ]
security-administrator@host:fips# set system login password format sha512
```

- Commit the configuration:

```
[edit]
security-administrator@host:fips# commit
```

  The new hash algorithm applies only to those passwords that you generate after committing this configuration.

  Your device supports ECDSA (P-256, P-384, and P-521) and RSA (2048, 3072, and 4096 modulus bit length) key types.

Configure Network Device Collaborative Protection Profile Authorized Administrator in FIPS Mode

An NDcPPv2.2e-authorized administrator must have all the permissions, including the ability to change the device configuration.

To configure an authorized administrator:

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

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

2. Configure the hashed algorithm for plain-text passwords as `sha512`.

```
[edit]
security-administrator@host:fips# set system login password format sha512
```

3. Commit the changes.

```
[edit]
security-administrator@host:fips# commit
```

4. Define the NDcPPv2.2e-authorized administrator.

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

Or

```
[edit]
security-administrator@host:fips# set system login user NDcPPv2-user class security-admin
authentication plain-text-password
```

5. Load an SSH key file that was previously generated using ssh-keygen. This command loads RSA (SSHv2), or ECDSA (SSHv2).

```
[edit]
security-administrator@host:fips# set system root-authentication load-key-file url:filename
```

6. Set the `log-key-changes` configuration statement to log all instances of addition or removal of SSH authentication keys.

```
[edit]
security-administrator@host:fips# set system services ssh log-key-changes
```

When you enable and commit the `log-key-changes` statement, Junos OS logs the changes to the set of authorized SSH keys for each user (including the added or removed keys). Junos OS logs the differences since the last time you enabled the `log-key-changes` configuration statement. If you never enabled the `log-key-changes` configuration statement, then Junos OS logs all the authorized SSH keys.

7. Commit the changes.

```
[edit]
security-administrator@host:fips# commit
```

For details on how to start with shell mode, see Overview for Junos OS Guide.

> **NOTE**: You must reset the root password when you change the sha256 or sha512 for the password storage format. This step protects the new password using the sha256 or sha512 hash algorithm. To reset the root password, use the `set system root-authentication` *`plain-text-password`* command, and confirm the new password when prompted.

Customize Time in FIPS Mode

To customize time, disable Network Time Protocol (NTP) and then set the date and time.

1. Disable NTP from the configuration mode.

```
[edit]
security-administrator@hostname:fips# deactivate groups global system ntp
security-administrator@hostname:fips# deactivate system ntp
security-administrator@hostname:fips# commit
security-administrator@hostname:fips# exit
```

2. Set the date and time from the operational mode. The date and time format is `YYYYMMDDHHMM.ss`.

```
security-administrator@hostname:fips> set date 201803202034.00
security-administrator@hostname:fips> set cli timestamp
```

Configure Inactivity Timeout Period, and Local and Remote Idle Session Termination

IN THIS SECTION

Inactivity timeout period configuration refers to the settings that determine how long your device waits while detecting no user activity before automatically logging out the user and locking the session. Inactivity timeout period is a security feature.

Configure Session Termination

Terminate the session after the security administrator specifies the inactivity timeout period.

1. Set the idle timeout.

```
[edit]
security-administrator@host:fips# set system login class security-admin idle-timeout 2
```

2. Configure the login access privileges.

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

3. Commit the configuration.

```
[edit]
security-administrator@host:fips# commit
```

```
commit complete
```

4. Set the password for the user.

```
[edit]
security-administrator@host:fips# set system login user NDcPPv2-user authentication plain-text-password
New password:
Retype new password:
```

5. Define a login class for the user.

```
[edit]
security-administrator@host:fips# set system login user NDcPPv2-user class security-admin
```

6. Commit the configuration.

```
[edit]
security-administrator@host:fips# commit

commit complete
```

Sample Output for Local Administrative Session Termination

Local administrative session termination refers to the process of ending a session that was initiated for administrative purposes on a local system.

```
con host
Trying a.b.c.d...
'autologin': unknown argument ('set ?' for help).
Connected to device.example.com
Escape character is '^]'.

Type the hot key to suspend the connection: <CTRL>Z
FreeBSD/amd64 (host) (ttyu0)
login: NDcPPv2-user
Password:
```

```
Last login: Sun Jun 23 22:42:27 from 10.224.33.70

--- JUNOS 22.4R2.8 Kernel 64-bit  JNPR-12.1-20230321.be5f9c0_buil
NDcPPv2-user@host> Warning: session will be closed in 1 minute if there is no activity
Warning: session will be closed in 10 seconds if there is no activity
Idle timeout exceeded: closing session

FreeBSD/amd64 (host) (ttyu0)
```

Sample Output for Remote Administrative Session Termination

Remote administrative session termination refers to the process of ending or disconnecting an administrative session that is being conducted remotely.

```
ssh NDcPPv2-user@host
Password:
Last login: Sun Jun 23 22:48:05 2019
--- JUNOS 22.4R2.8 Kernel 64-bit  JNPR-12.1-20230321.be5f9c0_buil
NDcPPv2-user@host> exit

Connection to host closed.
ssh NDcPPv2-user@host
Password:
Last login: Sun Jun 23 22:50:50 2019 from 10.224.33.70
--- JUNOS 22.4R2.8 Kernel 64-bit  JNPR-12.1-20230321.be5f9c0_buil
NDcPPv2-user@host> Warning: session will be closed in 1 minute if there is no activity
Warning: session will be closed in 10 seconds if there is no activity
Idle timeout exceeded: closing session

Connection to host closed.
```

Sample Output for User-Initiated Session Termination

User-initiated session termination refers to the process where a user takes the action to end a session.

```
ssh NDcPPv2-user@host
Password:
Last login: Sun Jun 23 22:48:05 2019
--- JUNOS 22.4R2.8 Kernel 64-bit  JNPR-12.1-20230321.be5f9c0_buil
NDcPPv2-user@host> exit

Connection to host closed.
```

Configure System Login Message and Announcement

A system login message appears before the user logs in, and a system login announcement appears after the user logs in. By default, the system does not display any login message or announcement.

You can define the text for the system login message and system login announcement that the device should display. If the message text contains spaces, enclose it in quotation marks. You can also format the message using the following special characters:

- **\n**—New line
- **\t**—Horizontal tab
- **\'**—Single quotation mark
- **\"**—Double quotation mark
- \\—Backslash

1. Configure a system login message through the console or management interface:

   ```
   [edit]
   security-administrator@host:fips# set system login message login-message-banner-text
   ```

2. Configure system announcement:

```
[edit]
security-administrator@host:fips# set system login announcement system-announcement-text
```

3. Commit the configuration:

```
[edit]
security-administrator@host:fips# commit
```

5

CHAPTER

Configure SSH in FIPS Mode

IN THIS CHAPTER

Configure SSH in FIPS Mode

SUMMARY

Learn to configure SSH in FIPS mode on your device to enable secure remote management.

IN THIS SECTION

You must configure SSH on your device in FIPS mode to ensure compliance with stringent security requirements. You can remotely log in to your device in FIPS mode using SSH.

Configure SSH on Your Device in FIPS Mode

You need to configure the following cryptographic algorithms on your device to validate SSH support for Collaborative Protection Profile for Network Devices (NDcPP).

1. Specify the permissible SSH host-key algorithms for the system services.

```
[edit]
security-administrator@host:fips# set system services ssh hostkey-algorithm ssh-ecdsa
security-administrator@host:fips# set system services ssh hostkey-algorithm no-ssh-dss
security-administrator@host:fips# set system services ssh hostkey-algorithm ssh-rsa
security-administrator@host:fips# set system services ssh hostkey-algorithm no-ssh-ed25519
```

2. Specify the SSH key exchange for Diffie-Hellman keys for the system services.

```
[edit]
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

3. Specify all the permissible message authentication code algorithms for SSHv2.

   ```
   [edit]
   security-administrator@host:fips# set system services ssh macs hmac-sha1
   security-administrator@host:fips# set system services ssh macs hmac-sha2-256
   security-administrator@host:fips# set system services ssh macs hmac-sha2-512
   ```

4. Specify the ciphers allowed for SSHv2.

   ```
   [edit]
   security-administrator@host:fips# set system services ssh ciphers aes128-cbc
   security-administrator@host:fips# set system services ssh ciphers aes256-cbc
   security-administrator@host:fips# set system services ssh ciphers aes128-ctr
   security-administrator@host:fips# set system services ssh ciphers aes256-ctr
   ```

5. Commit the changes:

   ```
   [edit]
   security-administrator@host:fips# commit
   ```

To disable the SSH service, deactivate and commit the SSH configuration:

```
[edit]
security-administrator@host:fips# deactivate system services ssh
```

To disable the Netconf service, deactivate and commit the Netconf configuration:

```
[edit]
security-administrator@host:fips# deactivate system services netconf ssh
```

See No Link Title for the details of cryptographic algorithms supported on your device.

Manage Remote User Login

IN THIS SECTION

An administrator might log in remotely to a device through SSH. The device locally stores the administrator credentials and provides access to the remote administrator if the system receives a valid username and password.

Invalid Login Attempts

If the administrator provides an invalid authentication credential, the device allows the administrator to reenter the credentials after an interval. The interval starts from 1 second after the first login attempt and increases exponentially after each attempt. If the number of authentication attempts exceeds the configured limit, the device does not accept the further authentication attempts for a configured lockout period. During this lockout period, the administrator can still have access to the device through the console as the root user. The administrator can reenter the credentials when the lockout period expires.

Configure Lockout Period

You can configure the duration for which the device gets locked after the administrator exceeds the limit for authentication attempts. The lockout period must be greater than zero. The configurable range for the lockout period is 1 through 43,200 minutes.

```
[edit system login]
security-administrator@host:fips# set retry-options lockout-period number
```

Configure Login Attempt Limit

You can configure the number of failed login attempts that an administrator can perform. The failed login attempts range is 1 through 10, with a default value of 10.

```
[edit system login]
security-administrator@host:fips# set retry-options tries-before-disconnect number
```

Configure Delay After Failed Login Attempts

You can configure a delay, in seconds, before a user can reenter a password after a failed attempt. The range is 1 through 3 seconds, with a default value of 2 seconds.

```
[edit system login]
security-administrator@host:fips# set retry-options backoff-threshold number
```

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

Use the `backoff-factor` option to specify the length of the delay in seconds:

```
[edit system login]
security-administrator@host:fips# set retry-options backoff-factor number
```

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

Configure SSH Root Login Denial

You can control user access through SSH. Configure the `ssh root-login deny` option to ensure that the root account remains active and continues to have local administrative privileges to the device even if other remote users are logged off.

```
[edit system]
security-administrator@host:fips# set services ssh root-login deny
```

SSHv2 Protocol Security

The SSHv2 protocol provides secure terminal sessions with secure encryption. The SSHv2 protocol enforces the task of running the key-exchange phase and changing the encryption and integrity keys for a session. The system performs key exchange periodically, after specified seconds or after specified bytes of data have passed over a connection. You can configure thresholds for SSH rekeying. The TSF ensures SSH connections use the same session keys for no longer than one hour. It also limits the session keys to 1 GB of transmitted data. The rekey must happen when either of the thresholds reaches its limit.

```
[edit system]
security-administrator@host:fips# set services ssh rekey time-limit number
```

The time limit before renegotiation of session keys is 1 through 1440 minutes.

```
[edit system]
security-administrator@host:fips# set services ssh rekey data-limit number
```

The data limit beyond which renegotiation happens for session keys is 51,200 through 4,294,967,295 byte.

NOTE: If the SSH connection breaks, you must reinitiate the SSH connection to log back into the device.

6

CHAPTER

MACsec in FIPS Mode

IN THIS CHAPTER

Media Access Control Security (MACsec) in FIPS Mode Overview

Media Access Control Security (MACsec) is an IEEE 802.1AE industry-standard security technology that provides secure communication for all traffic on Ethernet links. See IEEE 802.1AE standard details on the IEEE organization website at IEEE 802.1: BRIDGING & MANAGEMENT. MACsec provides point-to-point security on Ethernet links between the directly connected nodes. It is capable of identifying and preventing most security threats, including denial of service (DoS), intrusion, man-in-the-middle attacks, masquerading, passive wiretapping, and playback attacks.

With MACsec, you can secure point-to-point Ethernet links for almost all traffic, including frames from the following protocols:

- Link Layer Discovery Protocol (LLDP)
- Link Aggregation Control Protocol (LACP)
- Dynamic Host Configuration Protocol (DHCP)
- Address Resolution Protocol (ARP)
- Other protocols that are not typically secured on an Ethernet link because of limitations with other related security solutions.

You can use MACsec in combination with other security protocols such as IPsec and Secure Sockets Layer (SSL) to provide end-to-end network security.

A series of known-answer test (KAT) self-tests and crypto algorithms validations (CAV) validate each implementation of an algorithm. The following cryptographic algorithms are added specifically for MACsec.

- Advanced Encryption Standard (AES)—Cipher Message Authentication Code (CMAC)
- AES Key Wrap

A connectivity association (CA) is a set of devices that are authorized to communicate securely with each other using MACsec. Within a specific CA the connectivity association key (CAK), a cryptographic key, secures the communication. Within a network the connectivity association key name (CKN), a unique identifier, is used to distinguish different CAs. Configure a preshared key (PSK) on both ends of the communication link before the secure communication begins. This key is used to establish and authenticate the secure connection between devices.

Use the following PSK configurations for both CKN and CAK:

```
[edit]
security-administrator@host:fips# prompt security macsec connectivity-association connectivity-
association-name pre-shared-key cak
New cak (secret):
Retype new cak (secret):
```

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association ca_name pre-
shared-key ckn ckn
```

NOTE: In the above `set security macsec connectivity-association` *ca_name* `pre-shared-key ckn` *ckn* command, you need to provide a user-defined name for the *ca_name* variable option and a user-defined name in hexadecimal format for the *ckn* variable option.

The system exchanges a preshared key between directly connected links to establish a MACsec-secure link. The preshared key includes the CKN and the CAK. The CKN is a 64-digit hexadecimal number. The CAK is a 32-digit hexadecimal number. The CKN and CAK must match on both ends of a link to create a MACsec-secure link.

NOTE: To maximize security, we recommend you to configure all 64 digits of a CKN and all 32 digits of a CAK. If you do not configure all the digits for these keys, the system automatically configures all the remaining digits to 0. However, you receive a warning message when you attempt to commit the configuration.

After the successful exchange and verification of the preshared keys by both ends of the link, the MACsec Key Agreement (MKA) protocol establishes and manages the secure link. The MKA protocol then elects one of the two directly connected switches as the key server. The key server then shares a random security key with the other device over the MACsec-secure point-to-point link. The key server continues to periodically perform this action as long as MACsec is enabled.

For example, you can configure a CKN of `37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311` and CAK of `228ef255aa23ff6729ee664acb66e91f` on connectivity association.

RELATED DOCUMENTATION

Understanding Media Access Control Security (MACsec)

Configure MACsec in FIPS Mode

SUMMARY

Learn to configure MACsec in FIPS mode.

IN THIS SECTION

Configure MACsec on Your Device in FIPS Mode

You can configure MACsec to secure point-to-point Ethernet links by connecting your device with MACsec-capable Modular Interface Cards (MICs). You must separately configure each point-to-point Ethernet link that you want to secure with MACsec. You can enable MACsec on device-to-device links using static connectivity association key (CAK) security mode.

You can configure different interface rates such as 10 Gbps, 40 Gbps, and 100 Gbps in port mode and specific interface rates such as 10 Gbps, 40 Gbps, and 100 Gbps in PIC mode. In PIC mode, you can configure only one type of interface speed.

To configure MACsec on your device with Junos OS:

1. Customize time, see "Customize Time in FIPS Mode" on page 37.

   We don't claim NTP as part of FPT_STM_EXT.1 SFR. However, in this documentation, we provide the steps to activate or deactivate NTP services to validate MACsec tolerance and MACsec keychain.
2. Configure the MACsec security mode for the connectivity association.

```
[edit]
security-administrator@host:fips#  set security macsec connectivity-association connectivity-association-name exclude-protocol protocol-name
```

```
security-administrator@host:fips#  set security macsec connectivity-association connectivity-
association-name include-sci
security-administrator@host:fips#  set security macsec connectivity-association connectivity-
association-name mka key-server-priority priority-number
security-administrator@host:fips#  set security macsec connectivity-association connectivity-
association-name mka transmit-interval interval
security-administrator@host:fips#  set security macsec connectivity-association connectivity-
association-name offset 30
```

Based on your requirement, you can configure the `offset` *`offset-number`* value at the `set security macsec connectivity-association` *`connectivity-association-name`* hierarchy level as *`0`*, *`30`*, or *`50`*.

3. Create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

```
[edit]
security-administrator@host:fips# prompt security macsec connectivity-association
connectivity-association-name pre-shared-key cak
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name pre-shared-key ckn hexadecimal-number
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name replay-protect replay-window-size number-of-packets
```

Based on your requirement, you can configure the *number-of-packets* value at the `set security macsec connectivity-association` *`connectivity-association-name`* `replay-protect replay-window-size` hierarchy level to a value in the range of `0` through `65535`.

4. Set the MACsec Key Agreement (MKA) secure channel details.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name secure-channel secure-channel-name direction (inbound | outbound)
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name secure-channel secure-channel-name encryption (MACsec)
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name secure-channel secure-channel-name id mac-address mac-address
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name secure-channel secure-channel-name id port-id port-id-number
security-administrator@host:fips# set security macsec connectivity-association connectivity-
association-name secure-channel secure-channel-name offset (0|30|50)
security-administrator@host:fips# set security macsec connectivity-association connectivity-
```

```
association-name secure-channel secure-channel-name security-association security-association-
number key key-string
```

5. Set the MKA secure channel to security mode.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode security-mode
```

   Here, the `CA1` is an example of configured *connectivity-association-name*.

6. Assign a specified MACsec interface to the configured connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name connectivity-
association connectivity-association-name
```

Configure Static MACsec for Layer 3 Traffic

To configure static MACsec for Layer 3 traffic between two devices R0 and R1:

In R0:

1. Create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

```
[edit]
security-administrator@host:fips#  set security macsec connectivity-association CA1 pre-
shared-key ckn 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@host:fips#  prompt security macsec connectivity-association CA1 pre-
shared-key cak
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  set security macsec connectivity-association CA1 offset 30
```

2. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
```

```
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

3. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name traceoptions flag all
```

4. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-mode static-cak
```

5. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-server-priority 1
```

6. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka transmit-interval 3000
```

7. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-sci
```

8. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name connectivity-association CA1
```

```
security-administrator@host:fips# set interfaces interface-name unit 0 family inet address
10.1.1.1/24
```

In R1:

1. Create the preshared key by configuring the CKN and the CAK.

```
[edit]
security-administrator@host:fips#  set security macsec connectivity-association CA1 pre-
shared-key ckn 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@host:fips#  prompt security macsec connectivity-association CA1 pre-
shared-key cak
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  set security macsec connectivity-association CA1 offset 30
```

2. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

3. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name traceoptions
file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name traceoptions
flag all
```

4. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

5. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka
transmit-interval 3000
```

6. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-sci
```

7. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name connectivity-
association CA1
security-administrator@host:fips# set interfaces interface-name unit 0 family inet address
10.1.1.2/24
```

Configure MACsec with Keychain for Layer 3 Traffic

Synchronize both MACsec endpoint devices to NTP, as the time set for key start-time trigger must be the same on both the devices. To configure MACsec with keychain for Layer 3 traffic between devices R0 and R1:

In R0:

1. Assign a tolerance value to the authentication keychain.

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 tolerance 20
```

2. Create a secret password. It is a string of hexadecimal digits with up to 64 characters. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret data is used as a CAK.

You can configure up to 64 keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 start-time 2018-03-20.20:35
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 key-name 2345678922334455667788992223334445556667778889992222333344445552
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 start-time 2018-03-20.20:37
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 key-name 2345678922334455667788992223334445556667778889992222333344445553
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 start-time 2018-03-20.20:39
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 key-name 2345678922334455667788992223334445556667778889992222333344445554
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 start-time 2018-03-20.20:41
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 start-time 2018-03-20.20:43
```

Use the `prompt` command to enter a secret key value. For example, the secret key value can be set as *2345678922334455667788992223334123456789223344556677889922233341*.

You can configure up to 64 secret keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
```

```
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 4 secret
```

3. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-
suite gcm-aes-256
```

NOTE: The cipher value can also be set as **cipher-suite gcm-aes-128**.

4. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

6. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

7. Set the MKA key server priority.

   ```
   [edit]
   security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-server-priority 1
   ```

8. Set the MKA transmit interval.

   ```
   [edit]
   security-administrator@host:fips# set security macsec connectivity-association CA1 mka transmit-interval 3000
   ```

9. Enable the MKA secure channel.

   ```
   [edit]
   security-administrator@host:fips# set security macsec connectivity-association CA1 include-sci
   ```

10. Assign the connectivity association to an interface.

    ```
    [edit]
    security-administrator@host:fips# set security macsec interfaces interface-name connectivity-association CA1
    security-administrator@host:fips# set interfaces interface-name unit 0 family inet address 10.1.1.1/24
    ```

Configure MACsec with a keychain for Layer 3 traffic with the following steps.

In R1:

1. Assign a tolerance value to the authentication keychain.

   ```
   [edit]
   security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-kc1 tolerance 20
   ```

2. Create a secret password. It is a string of hexadecimal digits with up to 64 characters. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret data is used as a CAK.

You can configure up to 64 keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 key-name 234567892233445566778899222333444555666777888999222333344445551
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 start-time 2018-03-20.20:35
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 key-name 234567892233445566778899222333444555666777888999222333344445552
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 start-time 2018-03-20.20:37
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 key-name 234567892233445566778899222333444555666777888999222333344445553
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 start-time 2018-03-20.20:39
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 key-name 234567892233445566778899222333444555666777888999222333344445554
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 start-time 2018-03-20.20:41
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 key-name 234567892233445566778899222333444555666777888999222333344445555
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 start-time 2018-03-20.20:43
```

Use the `prompt` command to enter a secret key value. For example, the secret key value is: *2345678922334455667788992223334123456789223344556677889922233341*.

You can configure up to 64 secret keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
```

```
security-administrator@host:fips#  prompt security authentication-key-chains key-chain macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain macsec-kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
```

3. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-suite gcm-aes-256
```

4. **NOTE**:

   - You can use the non-XPN ciphers `AES-GCM-128` and `AES-GCM-256` for 10 Gbps or xe interfaces MACsec configuration only.

   - You can use the XPN ciphers `AES-GCM-XPN-128` and `AES-GCM-XPN-256` for 40 Gbps and 100 Gbps rates MACsec configuration. You can also use the XPN ciphers `AES-GCM-XPN-128` and `AES-GCM-XPN-256` for 10 Gbps or xe interfaces MACsec configuration, if it supports.

5. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

6. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
```

```
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

7. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

8. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-
server-priority 1
```

9. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka
transmit-interval 3000
```

10. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-
sci
```

11. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
connectivity-association CA1
security-administrator@host:fips# set interfaces interface-name unit 0 family inet address
10.1.1.2/24
```

Configure Static MACsec for Layer 2 Traffic

To configure static MACsec for Layer 2 traffic between the devices R0 and R1:

In R0:

1. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-server-priority 1
```

2. Create a secret password. It is a string of hexadecimal digits with up to 64 characters. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as a CAK.

```
[edit]
security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
```

   For example, the secret key value can be set as *2345678922334455667788992223334123456789223344556677889922233341*.

3. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-suite gcm-aes-256
```

4. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

6. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

7. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-
server-priority 1
```

8. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka
transmit-interval 3000
```

9. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-
sci
```

10. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
connectivity-association CA1
```

11. Configure VLAN tagging.

```
[edit]
security-administrator@host:fips# set interfaces interface-name1 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name1 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name1 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name1 unit 100 vlan-id 100
security-administrator@host:fips# set interfaces interface-name2 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name2 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name2 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name2 unit 100 vlan-id 100
```

12. Configure a bridge domain.

```
[edit]
security-administrator@host:fips# set bridge-domains BD-110 domain-type bridge
security-administrator@host:fips# set bridge-domains BD-110 vlan-id 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name1 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name2 100
```

The *interface-name1* and *interface-name2* options at the `set bridge-domains BD-110 interface` hierarchy level are user defined interfaces that are part of the bridge domain.

In R1:

1. Create the secret password to use. It is a string of hexadecimal digits with up to 64 characters. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret data is used as a CAK.

```
[edit]
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
```

For example, the secret key value can be set as `2345678922334455667788992223334123456789223344556677889922233341`.

2. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-
suite gcm-aes-256
```

3. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

4. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

5. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

6. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-
server-priority 1
```

7. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka
transmit-interval 3000
```

8. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-
sci
```

9. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
connectivity-association CA1
```

10. Configure VLAN tagging.

```
[edit]
security-administrator@host:fips# set interfaces interface-name1 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name1 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name1 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name1 unit 100 vlan-id 100
security-administrator@host:fips# set interfaces interface-name2 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name2 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name2 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name2 unit 100 vlan-id 100
```

11. Configure a bridge domain.

```
[edit]
security-administrator@host:fips# set bridge-domains BD-110 domain-type bridge
```

```
security-administrator@host:fips# set bridge-domains BD-110 vlan-id 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name1 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name2 100
```

Configure MACsec with Keychain for Layer 2 Traffic

Synchronize both MACsec endpoint devices to NTP, as the time set for key start time trigger must be the same on both the devices. To configure MACsec with keychain for Layer 3 traffic between the device R0 and R1:

In R0:

1. Assign a tolerance value to the authentication keychain.

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 tolerance 20
```

2. Create the secret password to use. It is a string of hexadecimal digits up to 64 characters long. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret data is used as a CAK.

   You can configure up to 64 keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 start-time 2018-03-20.20:35
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 key-name 2345678922334455667788992223334445556667778889992222333344445552
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 start-time 2018-03-20.20:37
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 key-name 2345678922334455667788992223334445556667778889992222333344445553
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 start-time 2018-03-20.20:39
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 key-name 2345678922334455667788992223334445556667778889992222333344445554
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
```

```
kc1 key 3 start-time 2018-03-20.20:41
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 start-time 2018-03-20.20:43
```

Use the `prompt` command to enter a secret key value. For example, the secret key value can be set as *2345678922334455667788992223334123456789223344556677889922233341*.

You can configure up to 64 secret keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
```

3. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-
suite gcm-aes-256
```

4. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

6. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

7. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-
server-priority 1
```

8. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka
transmit-interval 3000
```

9. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-
sci
```

10. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
connectivity-association CA1
```

11. Configure VLAN tagging.

```
[edit]
security-administrator@host:fips# set interfaces interface-name1 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name1 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name1 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name1 unit 100 vlan-id 100
security-administrator@host:fips# set interfaces interface-name2 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name2 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name2 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name2 unit 100 vlan-id 100
```

12. Configure a bridge domain.

```
[edit]
security-administrator@host:fips# set bridge-domains BD-110 domain-type bridge
security-administrator@host:fips# set bridge-domains BD-110 vlan-id 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name1 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name2 100
```

In R1:

1. Assign a tolerance value to the authentication keychain.

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 tolerance 20
```

2. Create a secret password. It is a string of hexadecimal digits with up to 64 characters. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as a CAK.

```
[edit]
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 key-name 23456789223344556677889922233344455566677788889992222333344445551
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 0 start-time 2018-03-20.20:35
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 key-name 23456789223344556677889922233344455566677788889992222333344445552
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 1 start-time 2018-03-20.20:37
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 key-name 23456789223344556677889922233344455566677788889992222333344445553
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 2 start-time 2018-03-20.20:39
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 key-name 23456789223344556677889922233344455566677788889992222333344445554
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 3 start-time 2018-03-20.20:41
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 key-name 23456789223344556677889922233344455566677788889992222333344445555
security-administrator@host:fips#  set security authentication-key-chains key-chain macsec-
kc1 key 4 start-time 2018-03-20.20:43
```

Use the `prompt` command to enter a secret key value. For example, you can set the secret key value as: *2345678922334455667788992223334123456789223344556677889922233341*.

You can configure up to 64 secret keys. See the following sample secret keys for reference:

```
[edit]
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
```

```
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@host:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
```

3. Associate the preshared keychain name with the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-
suite gcm-aes-256
```

4. Set the trace option values.

```
[edit]
security-administrator@host:fips# set security macsec traceoptions file MACsec.log
security-administrator@host:fips# set security macsec traceoptions file size 4000000000
security-administrator@host:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@host:fips# set security macsec interfaces interface-name
traceoptions flag all
```

6. Configure the MACsec security mode as `static-cak` for the connectivity association.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 security-mode static-cak
```

7. Set the MKA key server priority.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-server-priority 1
```

8. Set the MKA transmit interval.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 mka transmit-interval 3000
```

9. Enable the MKA secure channel.

```
[edit]
security-administrator@host:fips# set security macsec connectivity-association CA1 include-sci
```

10. Assign the connectivity association to an interface.

```
[edit]
security-administrator@host:fips# set security macsec interfaces interface-name connectivity-association CA1
```

11. Configure VLAN tagging.

```
[edit]
security-administrator@host:fips# set interfaces interface-name1 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name1 encapsulation flexible-ethernet-services
security-administrator@host:fips# set interfaces interface-name1 unit 100 encapsulation
```

```
vlan-bridge
security-administrator@host:fips# set interfaces interface-name1 unit 100 vlan-id 100
security-administrator@host:fips# set interfaces interface-name2 flexible-vlan-tagging
security-administrator@host:fips# set interfaces interface-name2 encapsulation flexible-
ethernet-services
security-administrator@host:fips# set interfaces interface-name2 unit 100 encapsulation
vlan-bridge
security-administrator@host:fips# set interfaces interface-name2 unit 100 vlan-id 100
```

12. Configure bridge domain.

```
[edit]
security-administrator@host:fips# set bridge-domains BD-110 domain-type bridge
security-administrator@host:fips# set bridge-domains BD-110 vlan-id 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name1 100
security-administrator@host:fips# set bridge-domains BD-110 interface interface-name2 100
```

Disable and Restart MACsec Sessions

Use the following configurations to disable and restart the MACsec sessions:

- To disable the MACsec session:

```
user@host# deactivate security macsec
```

- To restart the MACsec session:

```
user@host# run restart dot1x-protocol
```

Or

```
user@host# activate security macsec
```

7

CHAPTER

FIPS Self-Tests

IN THIS CHAPTER

Known Answer Test (KAT)

The cryptographic module enforces security rules to ensure that the Junos OS in FIPS mode meets the security requirements of FIPS 140-3 Level 1. To validate the output of cryptographic algorithms approved for FIPS and test the integrity of some system modules, the device performs the following series of KAT self-tests:

- `kernel_kats`—KAT for kernel cryptographic routines
- `macsec_kats`—KAT for MACsec cryptographic implementation
- `md_kats`—KAT for `libmd` and `libc`
- `openssl_kats`—KAT for OpenSSL cryptographic implementation
- `quicksec_kats`—KAT for QuickSec Toolkit cryptographic implementation

NOTE: You must run the LC KATs using AES-GCM-128 and AES-GCM-256 to meet MACsec requirements.

The device automatically performs the KAT self-tests at startup. The device also performs the conditional self-tests automatically to verify digitally signed software packages, generated random numbers, RSA and ECDSA keypairs, and manually entered keys.

After successful completion of the KATs, the device updates the log (syslog) file with the executed tests.

If one of the KATs fail, the device reboots continuously. You can reboot the device using a USB install media package.

The `file show /var/log/messages` command displays the system log.

```
root@host:fips> request system fips self-test

Testing kernel KATS:

  NIST 800-90 HMAC DRBG Known Answer Test:         Passed

  DES3-CBC Known Answer Test:                      Passed

  HMAC-SHA1 Known Answer Test:                     Passed

  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-384 Known Answer Test:                     Passed
  SHA-2-512 Known Answer Test:                     Passed
  AES128-CMAC Known Answer Test:                   Passed
  AES-CBC Known Answer Test:                       Passed
Testing MACSec KATS:
  AES128-CMAC Known Answer Test:                   Passed
  AES256-CMAC Known Answer Test:                   Passed
  AES-ECB Known Answer Test:                       Passed
  AES-KEYWRAP Known Answer Test:                   Passed
  KBKDF Known Answer Test:                         Passed
Testing libmd KATS:
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-512 Known Answer Test:                     Passed
Testing OpenSSL v1.0.2 KATS:
  FIPS ECDSA Known Answer Test:                    Passed
  FIPS ECDH Known Answer Test:                     Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  RSA-ENC Known Answer Test:                       Passed
  RSA-SIGN Known Answer Test:                      Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
  KDF-SSH-SHA256 Known Answer Test:                Passed
  KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed
  KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed
Testing OpenSSL KATS:
  FIPS ECDSA Known Answer Test:                    Passed
  FIPS ECDH Known Answer Test:                     Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  RSA-ENC Known Answer Test:                       Passed
```

```
    RSA-SIGN Known Answer Test:                      Passed

    KDF-IKE-V1 Known Answer Test:                    Passed

    KDF-SSH-SHA256 Known Answer Test:                Passed

    KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed

    KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed

  Testing QuickSec 7.0 KATS:

    DES3-CBC Known Answer Test:                      Passed

    HMAC-SHA1 Known Answer Test:                     Passed

    HMAC-SHA2-224 Known Answer Test:                 Passed

    HMAC-SHA2-256 Known Answer Test:                 Passed

    HMAC-SHA2-384 Known Answer Test:                 Passed

    HMAC-SHA2-512 Known Answer Test:                 Passed

    AES-CBC Known Answer Test:                       Passed

    AES-GCM Known Answer Test:                       Passed

    SSH-RSA-ENC Known Answer Test:                   Passed

    SSH-RSA-SIGN Known Answer Test:                  Passed

    SSH-ECDSA-SIGN Known Answer Test:                Passed

    KDF-IKE-V1 Known Answer Test:                    Passed

    KDF-IKE-V2 Known Answer Test:                    Passed

  Testing QuickSec KATS:

    DES3-CBC Known Answer Test:                      Passed

    HMAC-SHA1 Known Answer Test:                     Passed
    HMAC-SHA2-224 Known Answer Test:                 Passed

    HMAC-SHA2-256 Known Answer Test:                 Passed

    HMAC-SHA2-384 Known Answer Test:                 Passed

    HMAC-SHA2-512 Known Answer Test:                 Passed

    AES-CBC Known Answer Test:                       Passed

    AES-GCM Known Answer Test:                       Passed

    SSH-RSA-ENC Known Answer Test:                   Passed

    SSH-RSA-SIGN Known Answer Test:                  Passed

    KDF-IKE-V1 Known Answer Test:                    Passed

    KDF-IKE-V2 Known Answer Test:                    Passed

  Testing SSH IPsec KATS:

    NIST 800-90 HMAC DRBG Known Answer Test:         Passed

    DES3-CBC Known Answer Test:                      Passed

    HMAC-SHA1 Known Answer Test:                     Passed

    HMAC-SHA2-256 Known Answer Test:                 Passed

    AES-CBC Known Answer Test:                       Passed

    SSH-RSA-ENC Known Answer Test:                   Passed

    SSH-RSA-SIGN Known Answer Test:                  Passed

    KDF-IKE-V1 Known Answer Test:                    Passed

  Testing file integrity:

    File integrity Known Answer Test:                Passed
```

```
Testing crypto integrity:

  Crypto integrity Known Answer Test:              Passed

Expect an exec Authentication error...

/sbin/kats/run-tests: /sbin/kats/cannot-exec: Authentication error
```

NOTE: The device implements cryptographic libraries and algorithms that are not used in the FIPS-approved mode of operation.

8

CHAPTER

Monitor FIPS Mode

IN THIS CHAPTER

Configure Syslog Server on a Linux System for FIPS Mode

A secure Junos OS environment requires auditing of events and storing the events in a local audit file. The device simultaneously sends the recorded events to an external syslog server. The syslog server must have an SSH client with Network Configuration Protocol (NETCONF) support to receive the streamed syslog messages.

Use the below configuration details and establish a session between the target of evaluation (TOE) and the audit server. Track different devices actions to monitor the traffic that passes between the audit server and the device, and transfer the generated audit data to the audit server.

Ensure that the TOE summary specification (TSS) defines the method of transferring the audit data to the external audit server and the provision of the trusted channel.

The audit log required for Network Device Collaborative Protection Profile (NDcPP) compliance captures the following events:

- Committed changes
- System startup
- Login and logout of users
- Failure to establish an SSH session
- Establishment or termination of an SSH session
- Changes to the system time.
- Initiation of a system update

Configure event logging for a remote syslog server when the server initiates an SSH connection to the ToE.

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

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

   The system prompts you to enter the desired passphrase and displays the storage locations for the `syslog-monitor` keypair.

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

```
[edit system login]
security-administrator@host:fips# set class monitor permissions trace
```

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

```
[edit system login]
security-administrator@host:fips# set user syslog-mon class monitor authentication ssh-rsa
"public-key"
```

4. Set up NETCONF with SSH

```
[edit system services]
security-administrator@host:fips# set netconf ssh
```

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

```
[edit system]
security-administrator@host:fips# set syslog file messages any any
commit
```

6. On the remote syslog server, start the SSH agent `ssh-agent`. This step is required to manage the syslog-monitor key.

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

7. On the remote syslog server, add the `syslog-monitor` keypair to the `ssh-agent`.

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

You will be prompted to enter the desired passphrase. Enter the same passphrase that you entered in Step 1.

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

   ```
   security-administrator@host:fips# $ssh syslog-mon@NDcPP_TOE -s netconf > test.out
   ```

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

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

10. Monitor the event log received on the syslog server that the device generated for its admin actions. You can find examples of syslog messages below. Examine the traffic passing between the syslog server and the device to:

    - Ensure no one view the data while it passes between the audit server and the device.

    - Confirm the audit server successfully receives the data.

    Match the local event logs with the remote event logs on the syslog server. Record the details of the software (name and version) used on the syslog server during testing.

The following example shows the test log results for syslog server.

```
host@ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/host/.ssh/syslog-monitor.
Your public key has been saved in /home/host/.ssh/syslog-monitor.pub.
The key fingerprint is:
ef:75:d7:68:c5:ad:8d:6f:5e:7a:7e:9b:3d:f1:4d:3f syslog-monitor key pair
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|               ..|
|        S       +|
|         .     Bo|
|          . . *.X|
|         . . o E@|
|          .   .BX|
```

```
+-----------------+
[host@nms5-vm-linux2 ~]$ cat /home/host/.ssh/syslog-monitor.pub
ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCrUREJUBpjwAoIgRrGy9zgt+
D2pikk3Q/Wdf8I5vr+njeqJhCx2bUAkrRbYXNILQQAZbg7kLfi/8TqqL
eon4HOP2e6oCSorKdx/GrOTzLONL4fh0EyuSAk8bs5JuwWNBUokV025
gzpGFsBusGnlj6wqqJ/sjFsMmfxyCkbY+pUWb8m1/A9YjOFT+6esw+9S
tF6Gbg+VpbYYk/Oday4z+z7tQHRFSrxj2G92aoliVDBLJparEMBc8w
LdSUDxmgBTM2oadOmm+kreBUQjrmr6775RJn9H9YwIxKOxGm4SFnX/Vl4
R+lZ9RqmKH2wodIEM34K0wXEHzAzNZ01oLmaAVqT
syslog-monitor key pair
[host@nms5-vm-linux2 ~]$ eval `ssh-agent -s`
Agent pid 1453
[host@nms5-vm-linux2 ~]$ ssh-add ~/.ssh/syslog-monitor
Enter passphrase for /home/host/.ssh/syslog-monitor:
Identity added: /home/host/.ssh/syslog-monitor (/home/host/.ssh/syslog-monitor)
```

The following example shows test log results for net configuration channel

```
host@nms5-vm-linux2 ~]$ ssh syslog-mon@starfire -s netconf

 this is NDcPP test device

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

The following example shows the device-generated event logs that the syslog server receives.

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

The following example shows test log results for net configuration channel

```
host@nms5-vm-linux2 ~]$ ssh syslog-mon@starfire -s netconf
 this is NDcPP test device

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

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

```
Local :
an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Redundancy 
interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd', PID 4317, 
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Dynamic 
```

```
flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/dfcd', PID 4318,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/cfmd', PID 4319,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2ald', PID 4320,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2cpd'
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state machinesJan 20
17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2cpd', PID 4321,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Multicast
Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/mcsnoopd', PID
4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: commit
wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: start ffp
activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to profiles
...............
```

```
Remote :
an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Redundancy
interface management process checking new configuration
```

```
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd', PID 4317,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Dynamic
flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/dfcd', PID 4318,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/cfmd', PID 4319,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2ald', PID 4320,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Layer 2
Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/l2cpd'
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state machinesJan 20
17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/l2cpd', PID 4321,
status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: Multicast
Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/mcsnoopd', PID
4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: commit
wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress:
activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in progress: start ffp
activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to profiles
...............
```

Configure Audit Log Options in FIPS Mode

SUMMARY

Learn to configure audit log options on your device in FIPS mode.

IN THIS SECTION

Audit log options refer to the settings and configurations available for managing and maintaining audit logs in a system. Audit logs are records that document various activities, changes to the configuration, and events that occur within a system. Audit logs provide a trail that can be used for security, compliance, and troubleshooting.

Configure Audit Log Options on Your Device in FIPS Mode

To configure audit log options on your device in FIPS mode:

1. Specify the number of files you need to archive in the system logging facility.

```
[edit system syslog]
security-administrator@host:fips# set archive files 2
```

2. Specify the file in which to log data.

```
[edit system syslog]
security-administrator@host:fips# set file syslog any any
```

3. Specify the size of files you need to archive.

```
[edit system syslog]
security-administrator@host:fips# set file syslog archive size 10000000
```

4. Specify the priority and facility in messages for the system logging facility.

   ```
   [edit system syslog]
   security-administrator@host:fips# set file syslog explicit-priority
   ```

5. Configure system message logging to follow a structured format.

   ```
   [edit system syslog]
   security-administrator@host:fips# set file syslog structured-data
   ```

6. Commit the changes:

   ```
   [edit]
   security-administrator@host:fips# commit
   ```

Sample Code Audits for Configuration Change Audit

The following sample code audits all changes to the configured secret data and sends the logs to a file named **Audit-File**.

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

This sample code expands the minimum audit scope from the changes to the secret data to all the changes to the configuration and sends the logs to a file named **Audit-File**.

```
[edit system]
syslog {
    file Audit-File {
        any any;
        authorization info;
```

```
        change-log any;
        interactive-commands info;
        kernel info;
        pfe info;
    }
}
```

Example: The System Logging for Configuration Changes

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

```
[edit system]
location {
    country-code US;
    building B1;
}
...
login {
    message "UNAUTHORIZED USE OF THIS ROUTER\n\tIS STRICTLY PROHIBITED!";
        user admin {
            uid 2000;
             class super-user;
        authentication {
            encrypted-password “$ABC123”;
               # SECRET-DATA
        }
    }
}
radius-server 192.0.2.15 {
    secret “$ABC123” # SECRET-DATA
}
services {
    ssh;
}
syslog {
    user *{
        any emergency;
    }
    file messages {
        any notice;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
...
...
```

The following example shows configuration statements that change the secret data and add a new user.

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

Event Log in FIPS Mode

SUMMARY

Learn to analyze the event logs in FIPS mode.

IN THIS SECTION

Event Log in FIPS Mode Overview

An event log is a detailed record of security-related event that occur within an information system. Event logs capture a variety of information, such as system messages, security events, application events, and user activities. Event logs are critical for monitoring, diagnosing, and troubleshooting issues, as well as for ensuring security and compliance.

The evaluated configuration requires the audit of configuration changes through the system log. In addition, Junos OS can:

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

The log for the evaluated configuration must capture the system events. Table 11 on page 96 shows samples of syslog auditing for NDcPPv2.2e:

Table 11: Auditable Events

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FAU_GEN.1	None	None	—
FAU_GEN.2	None	None	—
FAU_STG_EXT.1	None	None	—
FAU_STG.1	None	None	—

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FCS_CKM.1	None	None	—
FCS_CKM.2	None	None	—
FCS_CKM.4	None	None	—
FCS_COP.1/ DataEncryption	None	None	—
FCS_COP.1/SigGen	None	None	—
FCS_COP.1/Hash	None	None	—
FCS_COP.1/ KeyedHash	None	None	—
FCS_RBG_EXT.1	None	None	—
FDP_RIP.2	None	None	—

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated

FIA_AFL.1	Unsuccessful login attempts limit is reached or exceeded.	Origin of the attempt (for example, IP address)	sshd - SSHD_LOGIN_ATTEMPTS_THRESHOLD: Threshold for unsuccessful authentication attempts (3) reached by user 'security-administrator'
			Login lockout configuration details:
			`[edit]`
			`root@host:fips# run show system login lockout`
			`User`
			`Lockout start`
			`Lockout end`
			`security-administrator   2023-01-10 15:03:26 IST    2023-01-10 15:04:26 IST`
			Log for the login lockout configuration:
			Jan 10 15:03:26 host sshd[63687]: LIBJNX_LOGIN_ACCOUNT_LOCKED: Account for user 'security-administrator' has been locked out from logins
			Status of the session closed after the lockout period:
			`ssh security-administrator@host`
			`Password:`
			`Connection closed by 10.209.21.170 port 22`
			Log for the closed session after lockout period:
			Jan 10 15:04:10 host sshd[63694]: PAM_USER_LOCK_ACCOUNT_LOCKED: Account for user security-administrator is locked.
			Establishes the session through the console as the root user during lockout period:

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			login: security-administrator Password: Last login: Tue Jan 10 15:01:43 on ttyu0 --- JUNOS 22.4R2.8 Kernel 64-bit JNPR-12.1-20230321.be5f9c0_buil security-administrator@bm-a:fips> [edit] root@host:fips# run show system users 3:04PM up 4 days, 3:59, 2 users, load averages: 0.28, 0.21, 0.22 USER TTY FROM LOGIN@ IDLE WHAT security-a u0 - 3:03PM - -cli (cli) Log for the session established through the console as the root user during lockout period: Jan 10 15:03:52 host login[63625]: LOGIN_INFORMATION: User security-administrator logged in from host [unknown] on device ttyu0
FIA_PMG_EXT.1	None	None	—

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (for example, IP address).	Successful Remote Login mgd 70652 UI_AUTH_EVENT [junos@2636.1.1.1.2.164 username="root" authentication-level="super-user"] Authenticated user 'root' assigned to class 'super-user' mgd 70652 UI_LOGIN_EVENT [junos@2636.1.1.1.2.164 username="root" class-name="super-user" local-peer="" pid="70652" ssh-connection="10.223.5.251 53476 10.204.134.54 22" client-mode="cli"] User 'root' login, class 'super-user' [70652], ssh-connection '10.223.5.251 53476 10.204.134.54 22', client-mode 'cli' Unsuccessful Remote Login sshd - SSHD_LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="10.223.5.251"] Login failed for user 'root' from host '10.223.5.251' Successful Local Login login 2671 LOGIN_INFORMATION [junos@2636.1.1.1.2.164 username="root" hostname="[unknown\]" tty-name="ttyu0"] User root logged in from host [unknown] on device ttyu0 login 2671 LOGIN_ROOT [junos@2636.1.1.1.2.164 username="root" hostname="[unknown\]" tty-name="ttyu0"] User root logged in as root from host [unknown] on device ttyu0 Unsuccessful Local Login login 70818 LOGIN_PAM_ERROR [junos@2636.1.1.1.2.164 username="root" error-message="error in service module"]

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			Failure while authenticating user root: error in service module login 70818 LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="ttyu0"] Login failed for user root from host ttyu0

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (for example, IP address).	Successful Remote Login mgd 70652 UI_AUTH_EVENT [junos@2636.1.1.1.2.164 username="root" authentication-level="super-user"] Authenticated user 'root' assigned to class 'super-user' mgd 70652 UI_LOGIN_EVENT [junos@2636.1.1.1.2.164 username="root" class-name="super-user" local-peer="" pid="70652" ssh-connection="10.223.5.251 53476 10.204.134.54 22" client-mode="cli"] User 'root' login, class 'super-user' [70652], ssh-connection '10.223.5.251 53476 10.204.134.54 22', client-mode 'cli' Unsuccessful Remote Login sshd - SSHD_LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="10.223.5.251"] Login failed for user 'root' from host '10.223.5.251' Successful Local Login login 2671 LOGIN_INFORMATION [junos@2636.1.1.1.2.164 username="root" hostname="[unknown\]" tty-name="ttyu0"] User root logged in from host [unknown] on device ttyu0 login 2671 LOGIN_ROOT [junos@2636.1.1.1.2.164 username="root" hostname="[unknown\]" tty-name="ttyu0"] User root logged in as root from host [unknown] on device ttyu0 Unsuccessful Local Login login 70818 LOGIN_PAM_ERROR [junos@2636.1.1.1.2.164 username="root" error-message="error in service module"]

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			Failure while authenticating user root: error in service module login 70818 LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="ttyu0"] Login failed for user root from host ttyu0
FIA_UAU.7	None	None	
FMT_MOF.1/ ManualUpdate	Any attempt to initiate a manual update.	None	UI_CMDLINE_READ_LINE [junos@2636.1.1.1.2.164 username="sec-officer" command="request vmhost software add junos-vmhost-install-mx-x86-64-22.4R1.10.tgz no-validate "] User 'sec-officer', command 'request vmhost software add junos-vmhost-install-mx-x86-64-22.4R1.10.tgz no-validate'
FMT_MTD.1/ CoreData	All management activities of TSF data.	None	Refer to the audit events listed in this table.
FMT_SMF.1/IPS	None	None	None
FMT_SMF.1/ND	None	None	None
FMT_SMR.2	None	None	—
FPT_SKP_EXT.1	None	None	—
FPT_APW_EXT.1	None	None	—

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FPT_TST_EXT.1 **NOTE**: If there is a self-test error, you can recover the device via USB recovery. If USB recovery fails, you can contact JTAC for support (https://support.juniper.net/support/).	None	None	Enter `request system fips self-test` at command line for on demand self-test. or Reboot the device to view the self-test during start-up.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None	UI_CMDLINE_READ_LINE [junos@2636.1.1.1.2.164 username="sec-officer" command="request vmhost software add junos-vmhost-install-mx-x86-64-22.4R1.10.tgz no-validate "] User 'sec-officer', command request vmhost software add junos-vmhost-install-mx-x86-64-22.4R1.10.tgz no-validate '
FPT_STM_EXT.1 **NOTE**: We don't claim Network Time Protocol (NTP) as a part of FPT_STM_EXT.1 SFR. However, as per this documentation, you can activate or deactivate NTP services to validate MACsec tolerance and MACsec keychain.	• Discontinuous changes to time. • Either administrator actuated or changed through an automated process.	• Discontinuous changes to time: The old and new values for the time. • Origin of the attempt to change time for success and failure (such as, IP address).	mgd 71079 UI_CMDLINE_READ_LINE [junos@2636.1.1.1.2.164 username="root" command="set date 202005201815.00 "] User 'root', command 'set date 202005201815.00 ' mgd 71079 UI_COMMIT_PROGRESS [junos@2636.1.1.1.2.164 message="signaling 'Network security daemon', pid 2641, signal 31, status 0 with notification errors enabled"] Commit operation in progress: signaling 'Network security daemon', pid 2641, signal 31, status 0 with notification errors enabled nsd 2641 NSD_SYS_TIME_CHANGE - System time has changed

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FTA_SSL_EXT.1 (if *terminate the session* is selected)	The termination of a local interactive session by the session-locking mechanism	None	cli - UI_CLI_IDLE_TIMEOUT [junos@2636.1.1.1.2.164 username="root"] Idle timeout for user 'root' exceeded and session terminated
FTA_SSL.3	The termination of a remote session by the session-locking mechanism	None	cli - UI_CLI_IDLE_TIMEOUT [junos@2636.1.1.1.2.164 username="root"] Idle timeout for user 'root' exceeded and session terminated
FTA_SSL.4	The termination of an interactive session	None	mgd 71668 UI_LOGOUT_EVENT [junos@2636.1.1.1.2.164 username="root"] User 'root' logout
FTA_TAB.1	None	None	—
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure	sshd 72404 - - Unable to negotiate with 1.1.1.2 port 42168: no matching cipher found. Their offer: chacha20-poly1305@openssh.com, aes128-ctr, aes192-ctr,aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, aes192-cbc, aes256-cbc

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FTP_ITC.1	• Initiation of a trusted channel. • Termination of a trusted channel. • Failure of a trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt	Initiation of the trusted path sshd 72418 - - Accepted keyboard-interactive/pam for root from 10.223.5.251 port 42482 ssh2 Termination of the trusted path sshd 72418 - - Disconnected from user root 10.223.5.251 port 42482 Failure of the trusted path sshd - SSHD_LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="10.223.5.251"] Login failed for user 'root' from host '10.223.5.251'
FTP_TRP.1/Admin	• Initiation of a trusted channel. • Termination of a trusted channel. • Failure of a trusted channel functions.	None	Initiation of the trusted path sshd 72418 - - Accepted keyboard-interactive/pam for root from 10.223.5.251 port 42482 ssh2 Termination of the trusted path sshd 72418 - - Disconnected from user root 10.223.5.251 port 42482 Failure of the trusted path sshd - SSHD_LOGIN_FAILED [junos@2636.1.1.1.2.164 username="root" source-address="10.223.5.251"] Login failed for user 'root' from host '10.223.5.251'

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure	sshd 72404 - - Unable to negotiate with 1.1.1.2 port 42168: no matching cipher found. Their offer: chacha20-poly1305@openssh.com, aes128-ctr,aes192-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, aes192-cbc, aes256-cbc
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure	verify-sig 72830 - - cannot validate ecerts.pem: subject issuer mismatch: /C=US/ ST=CA/L=Sunnyvale/O=Juniper Networks/ OU=Juniper CA/CN=PackageProduction TestEc_2017_NO_DEFECTS/emailAddress =ca@juniper.net
FIA_X509_EXT.2	None	None	—
FIA_X509_EXT.3	None	None	—
FMT_MOF.1/ Functions	Modification of the audit data transmission behavior to an external IT entity, the handling of audit data, the audit functionality when local audit storage space is full.	None	mgd 71891 UI_RESTART_EVENT [junos@2636.1.1.1.2.164 username="root" process-name="Network security daemon" description=" immediately"] User 'root' restarting daemon 'Network security daemon' immediately init - - - network-security (PID 72907) terminated by signal number 9! init - - - network-security (PID 72929) started
FMT_MOF.1/ Services	Starting and stopping of services	None	—

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FMT_MTD.1/ CryptoKeys	Management of cryptographic keys.	None	SSH key ssh-keygen 2706 - - Generated SSH key file /root/.ssh/id_rsa.pub with fingerprint SHA256:EQotXjlahhlVplg + YBLbFR3TdmJMpm6D1FSjRo6lVE4 ssh-keygen 2714 - - Generated SSH key file /root/.ssh/id_ecdsa.pub with fingerprint SHA256:ubQWoesME9bpOT1e/ sYv871hwWUzSG8hNqyMUe1cNc0 IPSEC keys pkid 2458 PKID_PV_KEYPAIR_GEN [junos@2636.1.1.1.2.164 argument1="384" argument2="ECDSA" argument3="cert1"] A 384 bit ECDSA key-Pair has been generated for cert1 pkid 2458 PKID_PV_KEYPAIR_GEN [junos@2636.1.1.1.2.164 argument1="4096" argument2="RSA" argument3="cert2"] A 4096 bit RSA key-Pair has been generated for cert2

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted or received during session establishment.	user@host:fips# run show log iked | no-more | grep vpn Jun 14 10:40:49.291712 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ipsec-sa selection successful for spi (0x8a45e874) local-ip (20.1.1.1) remote-ip (20.1.1.2) vpn (IPSEC_VPN) user@host:fips# run show log iked | no-more | grep success Jun 14 10:40:49.278061 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ike-atec-dh-generate successful response received for ipc-index=45109,local-ip=none,remote-ip=none Jun 14 10:40:49.290742 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] atec-validate-migrate for ed (0x2c09028) success in remote id validation Jun 14 10:40:49.291392 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] TSi: traffic-selector-match for ts-match Successful,C:ipv4(0.0.0.0-255.255.255.255) R:ipv4(10.1.1.0-10.1.1.255) N:ipv4(10.1.1.0-10.1.1.255) Jun 14 10:40:49.291656 [EXT] [TUNL] [20.1.1.1 <-> 20.1.1.2] ike_tunnel_anchor_node_tunnel_add: Anchor tunnel add for tunnel 500009: success total tunnel adds:9 Jun 14 10:40:49.291682 [DET] [TUNL] [20.1.1.1 <-> 20.1.1.2] tunnel-sadb-add success with local-spi (0x8a45e874) Jun 14 10:40:49.291712 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ipsec-sa selection successful for spi (0x8a45e874) local-ip

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			(20.1.1.1) remote-ip (20.1.1.2) vpn (IPSEC_VPN) Jun 14 10:40:49.292404 [TER] [PEER] [20.1.1.1 <-> 20.1.1.2] IKE: Gateway N:IKE_GW L:20.1.1.1:500 R:20.1.1.2:500 Successful ike-id:20.1.1.2 U:N/A IKE:IKEv2 Role:R Jun 14 10:40:49.294256 [DET] [DIST] [20.1.1.1 <-> 20.1.1.2] ike_dist_ipsec_tunnel_info_add: IPsec distribution tunnel info add to db successful Tunnel Id:500009 Client Id:20 Instance:0 Jun 14 10:40:49.295072 [EXT] [IPSC] [20.1.1.1 <-> 20.1.1.2] ipsec_common_msg_send: Successfully sent IPC msg tag 4 from iked to SPU.0.20 Jun 14 10:40:49.295292 [EXT] [IPSC] [20.1.1.1 <-> 20.1.1.2] ipsec_common_msg_send: Successfully sent IPC msg tag 4 from iked to SPU.0.21 Jun 14 10:40:49.296004 [DET] [STER] [20.1.1.1 <-> 20.1.1.2] Successfully modified st0 next hop meta data for tunnel 500009 Jun 14 10:40:49.297336 [EXT] [IPSC] [20.1.1.1 <-> 20.1.1.2] ipsec_common_msg_send: Successfully sent IPC msg tag 4 from iked to SPU.0.20 Jun 14 10:42:24.328902 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ike-atec-dh-generate successful response received for ipc-index=45111,local-ip=none,remote-ip=none Jun 14 10:42:24.332381 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ike-atec-dh-compute successful response received for ipc-index=0

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			Jun 14 10:42:24.333295 [DET] [PUBL] [20.1.1.1 <-> 20.1.1.2] publish-ike-sa successful for ike-sa-index 11282 ike-sa 0x21dec24 Jun 14 10:42:29.316880 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] TSi: traffic-selector-match for ts-match Successful,C:ipv4(0.0.0.0-255.255.255.255) R:ipv4(10.1.1.0-10.1.1.255) N:ipv4(10.1.1.0-10.1.1.255) Jun 14 10:42:29.316889 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] TSr: traffic-selector-match for ts-match Successful,C:ipv4(0.0.0.0-255.255.255.255) R:ipv4(30.1.1.0-30.1.1.255) N:ipv4(30.1.1.0-30.1.1.255) Jun 14 10:42:29.317147 [DET] [TUNL] [20.1.1.1 <-> 20.1.1.2] tunnel-sadb-add success with local-spi (0x80eeab18) Jun 14 10:42:29.317178 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ipsec-sa selection successful for spi (0x80eeab18) local-ip (20.1.1.1) remote-ip (20.1.1.2) vpn (IPSEC_VPN) Jun 14 10:42:29.320369 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ike-atec-dh-generate successful response received for ipc-index=45113,local-ip=none,remote-ip=none Jun 14 10:42:29.323800 [DET] [ATEC] [20.1.1.1 <-> 20.1.1.2] ike-atec-dh-compute successful response received for ipc-index=0 Jun 14 10:42:29.325513 [EXT] [IPSC] [20.1.1.1 <-> 20.1.1.2] ipsec_common_msg_send: Successfully sent IPC msg tag 4 from iked to SPU.0.20

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
FIA_X509_EXT.1	Session establishment with CA	Entire packet contents of packets transmitted or received during session establishment	kmd 7200 KMD_VPN_UP_ALARM_USER [junos@2636.1.1.1.2.164 vpn-name=""vpn1"" remote-address=""5.5.5.1"" local-address=""11.11.11.1"" ga teway-name=""gw1"" group-name=""vpn1"" tunnel-id=""131073"" interface-name=""st0.0"" internal-ip=""Not-Available"" name=""11.11.11.1"" peer-name=""5.5.5.1"" client-name=""Not-Applicable"" vrrp-group-id=""0"" traffic-selector-name= """" traffic-selector-cfg-local-id=""ipv4_subnet(any:0, [0..7\]=0.0.0.0/0)"" traffic-selector-cfg-remote-id= ""ipv4_subnet(any: 0, [0..7\]=0.0.0.0/0)"" argument1= ""Static""] VPN vpn1 from 5.5.5.1 is up. Local-ip: 11.11.11.1, gateway name: gw1, vpn name: vpn1, tunnel-id: 131073, local tunnel-if: st0.0, remote tunnel-ip: Not-Available, Local IKE-ID: 11.11.11.1, Remote IKE-ID: 5.5.5.1, AAA username: Not-Applicable, VR id: 0, Traffic-selector: , Traffic-selector local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Traffic-selector remote ID: ipv4_subnet(any:0, [0..7]=0.0.0.0/0), SA Type: Static

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated

FPF_RUL_EXT.1	Application of rules configured with the `log` operation	- Source and destination addresses.
		- Source and destination ports.
		- Transport layer protocol target of evaluation (TOE) interface
			`[edit]`
			`root@host:fips# run show firewall`
			
			`Filter: __default_bpdu_filter__`
			
			`Filter: fw_filter1`
			`Counters:`
			`Name`
			`         Bytes              Packets`
			`inc1`
			`             0                    0`
			`inc2`
			`           840                   10`
			
			`[edit]`
			`root@host:fips#`
			
			`[edit]`
			`root@host:fips# run show firewall log`
			`Log :`
			`Time      Filter    Action`
			`Interface           Protocol        Src`
			`Addr                         Dest Addr`
			`11:05:31  pfe       R`
			`st0.1               ICMP`
			`30.1.1.1                         10.1.1.1`
			`11:05:30  pfe       R`
			`st0.1               ICMP`
			`30.1.1.1                         10.1.1.1`
			`11:05:29  pfe       R`
			`st0.1               ICMP`
			`30.1.1.1                         10.1.1.1`
			`11:05:28  pfe       R`
			`st0.1               ICMP`
			`30.1.1.1                         10.1.1.1`
			
			`root@host:fips# run show firewall log`
			`Log :`
			`Time      Filter    Action`
			`Interface           Protocol        Src`

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated
			Addr Dest Addr 11:19:59 pfe R st0.1 TCP 30.1.1.1 10.1.1.1 root@host:fips# run show firewall log Log : Time Filter Action Interface Protocol Src Addr Dest Addr 13:00:18 pfe A ge-0/0/4.0 ICMP 30.1.1.5 10.1.1.1 13:00:17 pfe A ge-0/0/4.0 ICMP 30.1.1.5 10.1.1.1 13:00:16 pfe A ge-0/0/4.0 ICMP 30.1.1.5 10.1.1.1 13:00:15 pfe A ge-0/0/4.0 ICMP 30.1.1.5 10.1.1.1 root@host:fips# run show firewall log Log : Time Filter Action Interface Protocol Src Addr Dest Addr 13:00:45 pfe A ge-0/0/4.0 TCP 30.1.1.5 10.1.1.1

Table 11: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Records	How an Event Is Generated

	Indication of packets dropped due to high network traffic	TOE interface that is unable to process packets	`RT_FLOW - RT_FLOW_SESSION_DENY`
			`[junos@2636.1.1.1.2.164 sourceaddress="`
			`1.1.1. 2" source-port="10001"`
			`destination-address="2.2.2.2"`
			`destinationport="`
			`21" connection-tag="0" servicename="`
			`junos-ftp" protocol-id="6" icmptype="`
			`0" policy-name="p2" source-zone-na`
			`me="ZO_A" destination-zone-name="ZO_B"`
			`application="UNKNOWN" nestedapplication="`
			`UNKNOWN" username="N/A"`
			`roles="N/A" packet-incominginterface="`
			`ge-0/0/0.0" encrypted="No"`
			`reason="D enied by policy" sessionid-`
			`32="3" application-category="N/A"`
			`application-sub-category="N/A"`
			`applicationrisk="-`
			`1" application-characteristics="N/A"`
			`src-vrf-grp="N/A" dst-vrf-grp=" N/A"]`
			`session denied 1.1.1.2/10001->2.2.2.2/21`
			`0x0 junos-ftp 6(0) p2 ZO_A ZO_B`
			`UNKNOWN UNKNOWN N/A(N/A)`
			`ge-0/0/0.0 No Denied by policy 3 N/A N/A`
			`-1 N/A N/A N/A`

As a best practice, we recommend that you capture all changes to the configuration and remotely store the log information.

For more information about log details, see Specifying Log File Size, Number, and Archiving Properties

Interpret Event Messages

The following output shows a sample event message.

```
Feb 27 02:33:04  bm-a mgd[6520]: UI_LOGIN_EVENT: User 'security-officer' login, class 'j-super-
user' [6520], ssh-connection '', client-mode 'cli'
```

```
Feb 27 02:33:49  bm-a mgd[6520]: UI_DBASE_LOGIN_EVENT: User 'security-officer' entering
configuration mode
Feb 27 02:38:29  bm-a mgd[6520]: UI_CMDLINE_READ_LINE: User 'security-officer', command 'run
show log Audit_log | grep LOGIN
```

Table 12 on page 116 describes the fields for an event message. If the system logging utility cannot determine the value in a particular field, a hyphen (-) appears instead.

Table 12: Fields in Event Messages

Field	Description	Examples
timestamp	Time of message generation, in one of these formats: • *MMM-DD HH:MM:SS.MS+/-HH:MM* is the month, day, hour, minute, second and millisecond in local time. The hour and minute that follows the plus sign (+) or the minus sign (-) is the offset of the local time zone from UTC. • *YYYY-MM-DDTHH:MM:SS.MSZ* is the year, month, day, hour, minute, second, and millisecond in UTC.	Feb 27 02:33:04 is the timestamp expressed as local time in the United States. 2012-02-27T09:17:15.719Z is 2:33 AM UTC on 27 Feb 2012.
hostname	Name of the host that originally generates the message.	router1
process	Name of the Junos OS processes that generates the message.	mgd
processID	UNIX process ID (PID) of the Junos OS process that generates the message.	4153
TAG	The Junos OS system log message tag, which uniquely identifies the message.	UI_DBASE_LOGOUT_EVENT
username	Username of the user initiating the event	“admin”
message-text	English-language description of the event	set: [system radius-server 1.2.3.4 secret]

Log Changes to Secret Data

The following examples show audit logs of events that change the secret data. Whenever a configuration change happens, the syslog event captures logs similar to the following example:

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

Whenever a configuration update happens, the syslog event captures logs similar to the following example:

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

Login and Logout Events Using SSH

The system generates system log messages whenever a user successfully or unsuccessfully attempts to log in or log out using SSH. For example, the following logs show two failed log in attempts followed by a successful one and finally a log out event.

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

Logging of Audit Startup

The audit information logged includes instances of Junos OS startup. These logs identify the startup events of the audit system, which you cannot independently disable or enable. For example, when the Junos OS is restarts, the audit log contains the information similar to the following example:

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

9

CHAPTER

Disable FIPS Mode

IN THIS CHAPTER

Disable FIPS Mode

To begin repurposing your device for non-FIPS mode operation, perform zeroization on the device. See "Zeroize the System to Clear System Data for FIPS Mode" on page 10.

10

CHAPTER

Operational Commands

IN THIS CHAPTER

request vmhost zeroize no-forwarding

IN THIS SECTION

Syntax

```
request vmhost zeroize no-forwarding
```

Description

Remove all configuration information about the Routing Engines and reset all the key values. The command removes all data files, including customized configuration and log files, by unlinking the files from their directories. Additionally, the command removes all user-created files from the system, including all plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, IPsec, RADIUS, TACACS+, and SNMP.

This command reboots the device and sets it to the factory-default configuration. After the reboot, you cannot access the device through the management Ethernet interface. Log in through the console as the root user and start the Junos OS CLI by typing **cli** at the prompt.

Required Privilege Level

maintenance

Sample Output

request vmhost zeroize no-forwarding

```
user@host> request vmhost zeroize no-forwarding
VMHost Zeroization : Erase all data, including configuration and log files ?
[yes,no] (no) yes
warning: Vmhost will reboot and may not boot without configuration
warning: Proceeding with vmhost zeroize
Zeroise secondary internal disk ...
Proceeding with zeroize on secondary disk
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of secondary disk completed
Zeroize primary internal disk ...
Proceeding with zeroize on primary disk
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_rsa_key.pub
/etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_rsa_key
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of primary disk completed
Zeroize done
warning: Proceeding with vmhost reboot
Initiating vmhost reboot...
```

Release Information

Command introduced in Junos OS Release 15.1F3.