JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Common Criteria Evaluated Configuration Guide for EX4650-48Y, QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C Devices

Published
2026-02-17

RELEASE
22.3R1

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

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 Evaluated Configuration Guide for EX4650-48Y, QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C Devices
22.3R1
Copyright © 2026 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About This Guide | vi

1 Overview

Understanding the Common Criteria Evaluated Configuration | 2

Understanding Junos OS in FIPS Mode | 3

Understanding Common Criteria and FIPS Terminology and Supported Cryptographic Algorithms | 5

Identifying Secure Product Delivery | 9

Understanding Management Interfaces | 10

2 Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS in Common Criteria and FIPS | 13

Understanding the Operational Environment for Junos OS in FIPS Mode | 15

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 19

Downloading Software Packages from Juniper Networks | 21

Install Junos OS Software Package | 21

Understanding Zeroization to Clear System Data for FIPS Mode | 23

Zeroizing the System | 25

Enabling FIPS Mode | 26

Configuring Security Administrator and FIPS User Identification and Access | 31

- Configuring Security Administrator Login Access | 32
- Configuring FIPS User Login Access | 33

3 Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | 37

Configuring a Network Device collaborative Protection Profile for an Authorized Administrator | 39

Customize Time | 40

Configuring Inactivity Timeout Period Configuration, and Local and Remote Idle Session Termination | 41

Configure Session Termination | 41

Sample Output for Local Administrative Session Termination | 43

Sample Output for Remote Administrative Session Termination | 43

Sample Output for User Initiated Termination | 44

4 Configuring SSH and Console Connection

Configuring a System Login Message and Announcement | 46

Configuring SSH on the Evaluated Configuration | 47

Limiting the Number of User Login Attempts for SSH Sessions | 49

5 Configuring the Remote Syslog Server

Syslog Server Configuration on a Linux System | 53

Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server | 54

6 Configuring Audit Log Options

Configuring Audit Log Options in the Evaluated Configuration | 62

Configuring Audit Log Options | 62

Sample Code Audits of Configuration Changes | 63

7 Configuring Event Logging

Event Logging Overview | 83

Configuring Event Logging to a Local File | 84

Interpreting Event Messages | 84

Logging Changes to Secret Data | 86

Login and Logout Events Using SSH | 86

Logging of Audit Startup | 87

8 Configure MACsec on QFX5120-48YM

Overview of Media Access Control Security (MACsec) in FIPS mode | 89

Configure MACsec | 91

Customizing Time | 91

Configuring MACsec on a Device Running Junos OS | 92

Configuring Static MACsec with Layer 3 Traffic | 93

Configuring MACsec with keychain using Layer 3 Traffic | 97

Configuring Static MACsec for Layer 2 Traffic | 103

Configuring MACsec with keychain for Layer 2 Traffic | 108

Disable and Restart MACsec Sessions | 115

9 Performing Self-Tests on a Device

Understanding FIPS Self-Tests | 118

10 Configuration Statements

fips | 123

level | 124

11 Operational Commands

request system zeroize | 127

About This Guide

Use this guide to configure and evaluate EX4650-48Y, QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C devices for Common Criteria (CC) compliance. Common Criteria for information technology is an international agreement signed by several countries that permit the evaluation of security products against a common set of standards.

RELATED DOCUMENTATION

- Common Criteria and FIPS Certifications

CHAPTER 1

Overview

IN THIS CHAPTER

- Understanding the Common Criteria Evaluated Configuration | **2**
- Understanding Junos OS in FIPS Mode | **3**
- Understanding Common Criteria and FIPS Terminology and Supported Cryptographic Algorithms | **5**
- Identifying Secure Product Delivery | **9**
- Understanding Management Interfaces | **10**

Understanding the Common Criteria Evaluated Configuration

IN THIS SECTION

- Understanding Common Criteria | 2
- Supported Platforms | 3

This document describes the steps required to configure the device running Junos OS when the device is evaluated. This is referred to as the evaluated configuration. The following list describes the standards to which the EX4650-48Y, QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C devices have been evaluated:

- NDcPP v2.2e—https://www.niap-ccevs.org/MMO/PP/CPP_ND_V2.2E.pdf

The QFX5120-48YM is additionally evaluated to:

- MACsec EP v1.2—https://www.niap-ccevs.org/MMO/PP/pp_ndcpp_macsec_ep_v1.2.pdf

NOTE: The EX4650-48Y, QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C devices with Junos OS Release 22.3R1 is certified for Common Criteria with FIPS mode enabled on the devices.

For regulatory compliance information about Common Criteria, and FIPS for Juniper Networks products, see the Juniper Networks Compliance Advisor.

Understanding Common Criteria

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

evaluations are performed using a common methodology for information technology security evaluation.

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

Target of Evaluation (TOE) is a device or a system subjected to evaluation based on the Collaborative Protection Profile (cPP).

Supported Platforms

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

- EX4650-48Y (https://www.juniper.net/us/en/products/switches/ex-series/ex4650-campus-aggregation-and-core-switch.html)
- QFX5120-32C, QFX5120-48T, QFX5120-48Y, and QFX5120-48YM (https://www.juniper.net/us/en/products/switches/qfx-series/qfx5120-data-center-switches.html)
- QFX5110-32Q and QFX5110-48S (https://www.juniper.net/us/en/products/switches/qfx-series/qfx5110-ethernet-switch.html)
- QFX5200-32C and QFX5200-48Y (https://www.juniper.net/us/en/products/switches/qfx-series/qfx5200-data-center-switches.html)
- QFX5210-64C (https://www.juniper.net/us/en/products/switches/qfx-series/qfx5210-data-center-switches.html)

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 9

Understanding Junos OS in FIPS Mode

IN THIS SECTION

- About the Cryptographic Boundary on Your Device | 4
- How FIPS Mode Differs from Non-FIPS Mode | 4

Validated Version of Junos OS in FIPS Mode | 5

Federal Information Processing Standards (FIPS) 140-3 defines security levels for hardware and software that perform cryptographic functions.

Operating your device in a FIPS 140-3 Level 1 environment requires enabling and configuring FIPS mode on the device from the Junos OS CLI.

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

About the Cryptographic Boundary on Your Device

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



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

How FIPS Mode Differs from Non-FIPS Mode

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

- Self-tests of all cryptographic algorithms are performed at startup.
- Self-tests of random number and key generation are performed continuously.
- Weak cryptographic algorithms such as Data Encryption Standard (DES) and Message Digest 5 (MD5) are disabled.

- Weak or unencrypted management connections must not be configured. However, TOE allows local and un-encrypted console access across all modes of operation.
- Passwords must be encrypted with strong one-way algorithms that do not permit decryption.
- Junos-FIPS administrator passwords must be at least 10 characters long.
- Cryptographic keys must be encrypted before transmission.

Validated Version of Junos OS in FIPS Mode

To determine whether a Junos OS release is NIST-validated, see the compliance page on the Juniper Networks Web site (https://apps.juniper.net/compliance/).

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 9

Understanding Common Criteria and FIPS Terminology and Supported Cryptographic Algorithms

IN THIS SECTION

Terminology | 6
Supported Cryptographic Algorithms | 7

Use the definitions of Common Criteria and FIPS terms, and supported algorithms to help you understand Junos OS.

Terminology

Common Criteria	Common Criteria for information technology is an international agreement signed by several countries that permits the evaluation of security products against a common set of standards.
Security Administrator	For Common Criteria, user accounts in the TOE have the following attributes: user identity (user name), authentication data (password), and role (privilege). The Security Administrator is associated with the defined login class “security-admin”, which has the necessary permission set to permit the administrator to perform all tasks necessary to manage the Junos OS.
NDcPPv2.2e	Collaborative Protection Profile for Network Devices, Version 2.2e, dated 23 March 2020.
Critical security parameter (CSP)	Security-related information—for example, secret and private cryptographic keys and authentication data such as passwords and personal identification numbers (PINs)—whose disclosure or modification can compromise the security of a cryptographic module or the information it protects. For details, see "Understanding the Operational Environment for Junos OS in FIPS Mode" on page 15.
Cryptographic module	The set of hardware, software, and firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary. For fixed-configuration devices, the cryptographic module is the device case. For modular devices, the cryptographic module is the Routing Engine.
ESP	Encapsulating Security Payload (ESP) protocol. The part of the IPsec protocol that guarantees the confidentiality of packets through encryption. The protocol ensures that if an ESP packet is successfully decrypted, and no other party knows the secret key the peers share, the packet was not wiretapped in transit.
FIPS	Federal Information Processing Standards. FIPS 140-3 specifies requirements for security and cryptographic modules. Junos OS in FIPS mode complies with FIPS 140-3 Level 1.
FIPS maintenance role	The role the Security Administrator assumes to perform physical maintenance or logical maintenance services such as hardware or software diagnostics. For FIPS 140-3 compliance, the Security Administrator zeroizes the Routing Engine on entry to and exit from the FIPS maintenance role to erase all plain-text secret and private keys and unprotected CSPs.

> **NOTE**: The FIPS maintenance role is not supported on Junos OS in FIPS mode.

Hashing A message authentication method that applies a cryptographic technique iteratively to a message of arbitrary length and produces a hash *message digest* or *signature* of fixed length that is appended to the message when sent.

KATs Known answer tests. System self-tests that validate the output of cryptographic algorithms approved for FIPS and test the integrity of some Junos OS modules. For details, see "Understanding FIPS Self-Tests" on page 118.

SA Security association (SA). A connection between hosts that allows them to communicate securely by defining, for example, how they exchange private keys. As Security Administrator, you must manually configure an internal SA on devices running Junos OS in FIPS mode. All values, including the keys, must be statically specified in the configuration. On the devices with more than one Routing Engine, the configuration must match on both ends of the connection between the Routing Engines. For communication to take place, each Routing Engine must have the same configured options, which need no negotiation and do not expire. .

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

Zeroization Erasure of all CSPs and other user-created data on a device before its operation as a FIPS cryptographic module—or in preparation for repurposing the device for non-FIPS operation. The Security Administrator can zeroize the system with a CLI operational command. For details, see "Understanding Zeroization to Clear System Data for FIPS Mode" on page 23.

Supported Cryptographic Algorithms

Table 1 on page 8 summarizes the high level protocol algorithm support.

Table 1: Protocols Allowed in FIPS Mode

Protocol	Key Exchange	Authentication	Cipher	Integrity
SSHv2	• dh-group14-sha1 • ECDH-sha2-nistp256 • ECDH-sha2-nistp384 • ECDH-sha2-nistp521	• 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

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

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

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

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

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

HMAC	Defined as “Keyed-Hashing for Message Authentication” in RFC 2104, HMAC combines hashing algorithms with cryptographic keys for message authentication.
SHA-256, SHA-384, and SHA-512	Secure hash algorithms (SHA) belonging to the SHA-2 standard defined in FIPS PUB 180-2. Developed by NIST, SHA-256 produces a 256-bit hash digest, SHA-384 produces a 384-bit hash digest, and SHA-512 produces a 512-bit hash digest.
AES-CMAC	AES-CMAC provides stronger assurance of data integrity than a checksum or an error-detecting code. The verification of a checksum or an error-detecting code detects only accidental modifications of the data, while CMAC is designed to detect intentional, unauthorized modifications of the data, as well as accidental modifications.

RELATED DOCUMENTATION

Understanding FIPS Self-Tests | 118

Understanding Zeroization to Clear System Data for FIPS Mode | 23

Identifying Secure Product Delivery

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

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

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

Additionally, there are several checks that can be performed to ensure that the customer has received a box sent by Juniper Networks and not a different company masquerading as Juniper Networks. The

customer should perform the following checks upon receipt of a device to verify the authenticity of the device:

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

RELATED DOCUMENTATION

Understanding the Common Criteria Evaluated Configuration | 2

Understanding Management Interfaces

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

- Local Management Interfaces—The RJ-45 console port on the rear panel of a device is configured as RS-232 data terminal equipment (DTE). You can use the command-line interface (CLI) over this port to configure the device from a terminal.

- Remote Management Protocols—The device can be remotely managed over any Ethernet interface. SSHv2 is the only permitted remote management protocol that can be used in the evaluated configuration. The remote management protocols J-Web and Telnet are not available for use on the device.

RELATED DOCUMENTATION

Understanding the Common Criteria Evaluated Configuration | 2

2

CHAPTER

Configuring Roles and Authentication Methods

IN THIS CHAPTER

- Understanding Roles and Services for Junos OS in Common Criteria and FIPS | **13**
- Understanding the Operational Environment for Junos OS in FIPS Mode | **15**
- Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | **19**
- Downloading Software Packages from Juniper Networks | **21**
- Install Junos OS Software Package | **21**
- Understanding Zeroization to Clear System Data for FIPS Mode | **23**
- Zeroizing the System | **25**
- Enabling FIPS Mode | **26**
- Configuring Security Administrator and FIPS User Identification and Access | **31**

Understanding Roles and Services for Junos OS in Common Criteria and FIPS

IN THIS SECTION

- Security Administrator Role and Responsibilities | 13
- FIPS User Role and Responsibilities | 14
- What Is Expected of All FIPS Users | 14

The Security Administrator is associated with the defined login class “security-admin”, which has the necessary permission set to allow the administrator to perform all tasks necessary to manage the Junos OS. Administrative users (Security Administrator) must provide unique identification and authentication data before any administrative access to the system is granted.

Security Administrator roles and responsibilities are as follows:

1. Security Administrator can administer locally and remotely.
2. Create, modify, and delete administrator accounts, including configuration of authentication failure parameters.
3. Re-enable an Administrator account.
4. Responsible for the configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the evaluated product.

The Juniper Networks Junos operating system (Junos OS) running in non-FIPS mode allows a wide range of capabilities for users, and authentication is identity-based.

Security Administrator performs all FIPS-mode-related configuration tasks and issue all statements and commands for Junos OS in FIPS mode.

Security Administrator Role and Responsibilities

The Security Administrator is the person responsible for enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode on a device. The Security Administrator securely installs Junos OS

on the device, enables FIPS mode, establishes keys and passwords for other users and software modules, and initializes the device before network connection.



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

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

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

- Set the initial root password. The length of the password should be at least 10 characters.
- Reset user passwords with FIPS-approved algorithms.
- Examine log and audit files for events of interest.
- Erase user-generated files, keys, and data by zeroizing the device.

FIPS User Role and Responsibilities

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

FIPS user can view status output but cannot reboot or zeroize the device.

What Is Expected of All FIPS Users

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

All FIPS users must:

- Keep all passwords confidential.
- Store devices and documentation in a secure area.
- Deploy devices in secure areas.
- Check audit files periodically.

- Conform to all other FIPS 140-3 security rules.
- Follow these guidelines:
  - Users are trusted.
  - Users abide by all security guidelines.
  - Users do not deliberately compromise security.
  - Users behave responsibly at all times.

RELATED DOCUMENTATION

Zeroizing the System | 25

Configuring Security Administrator and FIPS User Identification and Access | 31

Understanding the Operational Environment for Junos OS in FIPS Mode

IN THIS SECTION

Hardware Environment for Junos OS in FIPS Mode | 16

Software Environment for Junos OS in FIPS Mode | 16

Critical Security Parameters | 17

A Juniper Networks device running the Junos operating system (Junos OS) in FIPS mode forms a special type of hardware and software operational environment that is 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 that no critical security parameters (CSPs) can cross using plain text. Each hardware component of the device that requires a cryptographic boundary for FIPS 140-3 compliance is a separate cryptographic module. There are two types of hardware with cryptographic boundaries in Junos OS in FIPS mode: one for each Routing Engine and one for entire chassis.

Cryptographic methods are not a substitute for physical security. The hardware must be located in a secure physical environment. Users of all types must not reveal keys or passwords, or allow written records or notes to be seen by unauthorized personnel.

Software Environment for Junos OS in FIPS Mode

A Juniper Networks device running Junos OS in FIPS mode forms a special type of non-modifiable 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 distribution. When a device is in FIPS mode, it can run only Junos OS.

The Junos OS in FIPS mode software environment is established after the Security Administrator successfully enables FIPS mode on a device. The Junos OS image that includes FIPS mode is available on the Juniper Networks website and can be installed on a functioning device.

For FIPS 140-3 compliance, we recommend deleting all user-created files and data from (*zeroizing*) the system immediately after enabling FIPS mode.

Enabling FIPS mode disables many of the usual Junos OS protocols and services. In particular, 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 upgrading to Junos OS in FIPS mode must conform to Junos OS in FIPS mode specifications. Passwords must be between 10 and 20 characters in length and require the use of at least three of the five defined character sets (uppercase and lowercase letters, digits, punctuation marks, and keyboard characters, such as % and &, not included in the other four categories). Attempts to configure passwords that do not conform to these rules result in an error. All passwords and keys used to authenticate peers must be at least 10 characters in length, and in some cases the length must match the digest size.



NOTE: Do not attach the device to a network until you, the Security Administrator, complete the configuration from the local console connection.

For strict compliance, do not examine core and crash dump information on the local console in Junos OS in FIPS mode because some CSPs might be shown in plain text.

Critical Security Parameters

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

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

Table 2 on page 17 lists CSPs on the device running Junos OS.

Table 2: Critical Security Parameters

CSP	Description	Zeroization method	Use
SSH-2 private host key	ECDSA / RSA key used to identify the host, generated the first time SSH is configured.	Zeroize command.	Used to identify the host.

Table 2: Critical Security Parameters *(Continued)*

CSP	Description	Zeroization method	Use
SSH-2 session key	Session key used with SSHv2 and as a Diffie-Hellman private key. Encryption: AES-128, AES-256. MACs: HMAC-SHA-1, HMAC-SHA-2-256,HMAC-SHA2-512. Key exchange: dh-group14-sha1, ECDH-sha2-nistp256, ECDH-sha2-nistp384, and ECDH-sha2-nistp521.	Power cycle and terminate session.	Symmetric key used to encrypt data between host and client.
User authentication key	Hash of the user's password: SHA-256, SHA-512.	Zeroize command.	Used to authenticate a user to the cryptographic module.
Security Administrator authentication key	Hash of the Security Administrator's password: SHA-256, SHA-512.	Zeroize command.	Used to authenticate the Security Administrator to the cryptographic module.
HMAC DRBG seed	Seed for deterministic randon bit generator (DRBG).	Seed is not stored by the cryptographic module.	Used for seeding DRBG.
HMAC DRBG V value	The value (V) of output block length (outlen) in bits, which is updated each time another outlen bits of output are produced.	Power cycle.	A critical value of the internal state of DRBG.
HMAC DRBG key value	The current value of the outlen-bit key, 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 2: Critical Security Parameters *(Continued)*

CSP	Description	Zeroization method	Use
NDRNG entropy	Used as entropy input string to the HMAC DRBG.	Power cycle.	A critical value of the internal state of DRBG.

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

Local passwords are hashed with the secure hash algorithm SHA-256, or SHA-512. Password recovery is not possible in Junos OS in FIPS mode. Junos OS in FIPS mode cannot boot into single-user mode without the correct root password.

RELATED DOCUMENTATION

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 19

Understanding Zeroization to Clear System Data for FIPS Mode | 23

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode

Ensure that the device is in FIPS mode before you configure the Security Administrator or any users. All passwords established for users by the Security Administrator must conform to the following Junos OS in FIPS mode requirements. Attempts to configure passwords that do not conform to the following specifications result in an error.

- Length. The passwords must contain at least 10 characters.
- Character set requirements. Passwords must contain at least three of the following five defined character sets:
  - Uppercase letters
  - Lowercase letters
  - Digits

    - Punctuation marks
    - Keyboard characters not included in the other four sets—such as the percent sign (%) and the ampersand (&)
- Authentication requirements. All passwords and keys used to authenticate peers must contain at least 10 characters, and 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.

Guidelines for strong passwords. Strong, reusable passwords can be based on letters from a favorite phrase or word and then concatenated with other unrelated words, along with added digits and punctuation. In general, a strong password is:

- Easy to remember so that users are not tempted to write it down.
- Made up of mixed alphanumeric characters and punctuation. For FIPS compliance include at least one change of case, one or more digits, and one or more punctuation marks.
- Changed periodically.
- Not divulged to anyone.

Characteristics of weak passwords. Do not use the following weak passwords:

- Words that might be found in or exist as a permuted form in a system files such as **/etc/passwd**.
- The hostname of the system (always a first guess).
- Any word or phrase that appears in a dictionary or other well-known source, including dictionaries and thesauruses in languages other than English; works by classical or popular writers; or common words and phrases from sports, sayings, movies or television shows.
- Permutations on any of the above—for example, a dictionary word with letters replaced with digits (**root**) or with digits added to the end.
- Any machine-generated password. Algorithms reduce the search space of password-guessing programs and so must not be used.

RELATED DOCUMENTATION

Understanding the Operational Environment for Junos OS in FIPS Mode | **15**

Downloading 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 Web account and a valid support contract. To obtain an account, complete the registration form at the Juniper Networks website: https://userregistration.juniper.net/.

NOTE: For QFX5110-32Q, QFX5110-48S, QFX5120-32C, QFX5120-48T, QFX5120-48Y, QFX5120-48YM, QFX5200-32C, QFX5200-48Y, and QFX5210-64C devices, FIPS is supported only on non-flex image. You have to upgrade to the non-flex image to enable FIPS mode.

To download software packages from Juniper Networks:

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

RELATED DOCUMENTATION

Install Junos OS Software Package | 21

Install Junos OS Software Package

You can use this procedure to upgrade Junos OS on the device with a single Routing Engine.

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

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

To install software upgrades on your device with a single Routing Engine:

1. Download the software package as described in "Downloading Software Packages from Juniper Networks " on page 21.
2. If you have not already done so, connect to the console port on the device from 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 *Junos OS Installation and Upgrade Guide* for instructions on performing this task.
4. (Optional) Copy the software package to the device. We recommend that you use FTP to copy the file to the **/var/tmp/** directory.

   This step is optional because Junos OS can also be upgraded when the software image is stored at a remote location. These instructions describe the software upgrade process for both scenarios.
5. Install the new package on the device:

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

   Replace ***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 *package* with the software package name.
     - **ftp://*hostname*/*pathname*/*package*.tgz**
     - **http://*hostname*/*pathname*/*package*.tgz**

   NOTE: If you need to terminate the installation, do not reboot your device; instead, finish the installation and then issue the `request system software delete package.tgz` command. This is your last chance to stop the installation.
6. Reboot the device to load the installation and start the new software:

   ```
   user@host> request system reboot
   ```

7. After the reboot has completed, log in and use the `show version` command to verify that the new version of the software is successfully installed.

```
user@host:fips> show version local
Hostname: hostname
Model: qfx5200-48y
Junos: 22.3R1.9
JUNOS OS Kernel 64-bit  [20220816.a81ed05_builder_stable_12_223]
JUNOS OS libs [20220816.a81ed05_builder_stable_12_223]
JUNOS OS runtime [20220816.a81ed05_builder_stable_12_223]
JUNOS OS time zone information [20220816.a81ed05_builder_stable_12_223]
JUNOS OS libs compat32 [20220816.a81ed05_builder_stable_12_223]
JUNOS OS 32-bit compatibility [20220816.a81ed05_builder_stable_12_223]
JUNOS py extensions [20220915.011905_builder_junos_223_r1]
JUNOS py base [20220915.011905_builder_junos_223_r1]
JUNOS OS vmguest [20220816.a81ed05_builder_stable_12_223]
JUNOS OS package [20220818.193938_builder_stable_12]
JUNOS OS crypto [20220816.a81ed05_builder_stable_12_223]
JUNOS OS boot-ve files [20220816.a81ed05_builder_stable_12_223]
JUNOS network stack and utilities [20220915.011905_builder_junos_223_r1]
JUNOS libs [20220915.011905_builder_junos_223_r1]
JUNOS libs compat32 [20220915.011905_builder_junos_223_r1]
JUNOS runtime [20220915.011905_builder_junos_223_r1]
JUNOS na telemetry [22.3R1.9]
JUNOS Web Management Platform Package [20220915.011905_builder_junos_223_r1]
...
```

Understanding Zeroization to Clear System Data for FIPS Mode

IN THIS SECTION

- Why Zeroize? | 24
- When to Zeroize? | 24

Zeroization completely erases all configuration information on the Routing Engines, including all plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, and IPsec.

The Security Administrator initiates the zeroization process by entering the **request system zeroize** operational command from the CLI after enabling FIPS mode. Use of this command is restricted to the Security Administrator.



CAUTION: Perform system zeroization with care. After the zeroization process is complete, no data is left on the Routing Engine. The device is returned to the factory default state, without any configured users or configuration files.

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

Why Zeroize?

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

When to Zeroize?

As Security Administrator, perform zeroization in the following situations:

- Before Enabling FIPS mode of operation: To prepare your device for operation as a FIPS cryptographic module, perform zeroization before enabling FIPS mode.
- Before repurposing to non-FIPS mode of operation: To begin repurposing your device for non-FIPS mode of operation, perform zeroization on the device.



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

RELATED DOCUMENTATION

Enabling FIPS Mode | 26

Zeroizing the System

Your device is not considered a valid FIPS cryptographic module until all critical security parameters (CSPs) have been entered—or reentered—while the device is in FIPS mode.

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

As Security Administrator, you run the `request system zeroize` command to remove all user-created files from a device and replace the user data with zeros. This command completely erases all configuration information on the Routing Engines, including all rollback configuration files and plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, and IPsec.

To zeroize your device:



CAUTION: Perform system zeroization with care. After the zeroization process is complete, no data is left on the Routing Engine. The device is returned to the factory default state, without any configured users or configuration files.

1. From the CLI, enter

```
root@host> request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files?   [yes, no] (no)
```

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

```
Erase all data, including configuration and log files?   [yes, no] (no) yes
warning: zeroizing localre
```

The entire operation can take considerable time depending on the size of the media, but all critical security parameters (CSPs) are removed within a few seconds. The physical environment must remain secure until the zeroization process is complete.

RELATED DOCUMENTATION

Enabling FIPS Mode | 26

Understanding Zeroization to Clear System Data for FIPS Mode | 23

Enabling FIPS Mode

FIPS mode is not automatically enabled when you install Junos OS on the device.

As Security Administrator, you must explicitly enable FIPS mode on the device by setting the FIPS level to 1 (one), the FIPS 140-3 level at which the devices are certified. A device on which FIPS mode is not enabled has a FIPS level of 0 (zero).



NOTE: To transition to FIPS mode, passwords must be encrypted with a FIPS-compliant hash algorithm. The encryption format must be SHA-256 or higher. Passwords that do not meet this requirement, such as passwords that are hashed with MD5, must be reconfigured or removed from the configuration before FIPS mode can be enabled.

To enable FIPS mode in Junos OS on the device:

1. The `fips-mode.tgz` is an optional package needed for enabling FIPS. This package is part of Junos OS software. To enable this package, use below command:

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

2. Enter configuration mode:

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

3. Enable FIPS mode on the device by setting the FIPS level to 1, and verify the level:

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

```
[edit]
root@host#show system
fips {
    level 1;
    }
```

4. Commit the configuration:

> **NOTE**: If the device terminal displays error messages about the presence of critical security parameters (CSPs), delete those CSPs, and then commit the configuration.

```
root@host# commit
configuration check succeeds
[edit]
  'system'
    reboot is required to transition to FIPS level 1
commit complete
```

5. Reboot the device:

```
[edit]
root@host# run request system reboot
Reboot the system ? [yes,no] (no) yes
```

During the reboot, the device runs Known Answer Tests (KATS). It returns a login prompt:

> **NOTE**: The new hash algorithm affect only those passwords that are generated after commit.

```
@ 1556787428 mgd start
Creating initial configuration: ...
```

```
mgd: Running FIPS Self-tests
mgd: Testing kernel KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   SHA-2-384 Known Answer Test:                     Passed
mgd:   SHA-2-512 Known Answer Test:                     Passed
mgd:   AES128-CMAC Known Answer Test:                   Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd: Testing MACSec KATS:
mgd:   AES128-CMAC Known Answer Test:                   Passed
mgd:   AES256-CMAC Known Answer Test:                   Passed
mgd:   AES-ECB Known Answer Test:                       Passed
mgd:   AES-KEYWRAP Known Answer Test:                   Passed
mgd:   KBKDF Known Answer Test:                         Passed
mgd: Testing libmd KATS:
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   SHA-2-512 Known Answer Test:                     Passed
mgd: Testing OpenSSL v1.0.2 KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   FIPS ECDSA Known Answer Test:                    Passed
mgd:   FIPS ECDH Known Answer Test:                     Passed
mgd:   FIPS RSA Known Answer Test:                      Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                 Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd:   AES-GCM Known Answer Test:                       Passed
mgd:   ECDSA-SIGN Known Answer Test:                    Passed
mgd:   KDF-IKE-V1 Known Answer Test:                    Passed
mgd:   KDF-SSH-SHA256 Known Answer Test:                Passed
mgd:   KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed
mgd:   KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed
mgd: Testing OpenSSL KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   FIPS ECDSA Known Answer Test:                    Passed
mgd:   FIPS ECDH Known Answer Test:                     Passed
mgd:   FIPS RSA Known Answer Test:                      Passed
```

```
mgd:   DES3-CBC Known Answer Test:                        Passed
mgd:   HMAC-SHA1 Known Answer Test:                       Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                   Passed
mgd:   AES-CBC Known Answer Test:                         Passed
mgd:   AES-GCM Known Answer Test:                         Passed
mgd:   ECDSA-SIGN Known Answer Test:                      Passed
mgd:   KDF-IKE-V1 Known Answer Test:                      Passed
mgd:   KDF-SSH-SHA256 Known Answer Test:                  Passed
mgd:   KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:      Passed
mgd:   KAS-FFC-EPHEM-NOKC Known Answer Test:              Passed
mgd: Testing QuickSec 7.0 KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:           Passed
mgd:   DES3-CBC Known Answer Test:                        Passed
mgd:   HMAC-SHA1 Known Answer Test:                       Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                   Passed
mgd:   AES-CBC Known Answer Test:                         Passed
mgd:   AES-GCM Known Answer Test:                         Passed
mgd:   SSH-RSA-ENC Known Answer Test:                     Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                    Passed
mgd:   SSH-ECDSA-SIGN Known Answer Test:                  Passed
mgd:   KDF-IKE-V1 Known Answer Test:                      Passed
mgd:   KDF-IKE-V2 Known Answer Test:                      Passed
mgd: Testing QuickSec KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:           Passed
mgd:   DES3-CBC Known Answer Test:                        Passed
mgd:   HMAC-SHA1 Known Answer Test:                       Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                   Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                   Passed
mgd:   AES-CBC Known Answer Test:                         Passed
mgd:   AES-GCM Known Answer Test:                         Passed
mgd:   SSH-RSA-ENC Known Answer Test:                     Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                    Passed
mgd:   KDF-IKE-V1 Known Answer Test:                      Passed
mgd:   KDF-IKE-V2 Known Answer Test:                      Passed
mgd: Testing SSH IPsec KATS:
```

```
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd:   SSH-RSA-ENC Known Answer Test:                   Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                  Passed
mgd:   KDF-IKE-V1 Known Answer Test:                    Passed
mgd: Testing file integrity:
mgd:   File integrity Known Answer Test:                Passed
mgd: Testing crypto integrity:
mgd:   Crypto integrity Known Answer Test:              Passed
```

Log in to the device. The CLI displays a banner that is followed by a prompt that includes ":fips":

```
--- JUNOS 22.3R1-20190716 built 2019-12-29 04:12:22 UTC
root@host:fips>
```

6. Reboot the device again to restore the HMAC-DRBG as an active random adapter:

```
[edit]
root@host# run request system reboot
Reboot the system ? [yes,no] (no) yes
```

During the reboot, the device runs Known Answer Tests (KATS) as shown in the step 4. It returns a login prompt:

```
--- JUNOS 22.3R1-20190716 built 2019-12-29 04:12:22 UTC
root@host:fips>
```

7. After the reboot has completed, log in and use the `show version local` command to verify.

```
user@host:fips> show version local
Hostname: hostname
Model: qfx5200-48y
Junos: 22.3R1.9
JUNOS OS Kernel 64-bit  [20220816.a81ed05_builder_stable_12_223]
JUNOS OS libs [20220816.a81ed05_builder_stable_12_223]
JUNOS OS runtime [20220816.a81ed05_builder_stable_12_223]
JUNOS OS time zone information [20220816.a81ed05_builder_stable_12_223]
```

```
JUNOS OS libs compat32 [20220816.a81ed05_builder_stable_12_223]
JUNOS OS 32-bit compatibility [20220816.a81ed05_builder_stable_12_223]
JUNOS py extensions [20220915.011905_builder_junos_223_r1]
JUNOS py base [20220915.011905_builder_junos_223_r1]
JUNOS OS vmguest [20220816.a81ed05_builder_stable_12_223]
JUNOS OS package [20220818.193938_builder_stable_12]
JUNOS OS crypto [20220816.a81ed05_builder_stable_12_223]
JUNOS OS boot-ve files [20220816.a81ed05_builder_stable_12_223]
JUNOS network stack and utilities [20220915.011905_builder_junos_223_r1]
JUNOS libs [20220915.011905_builder_junos_223_r1]
JUNOS libs compat32 [20220915.011905_builder_junos_223_r1]
JUNOS runtime [20220915.011905_builder_junos_223_r1]
JUNOS na telemetry [22.3R1.9]
JUNOS Web Management Platform Package [20220915.011905_builder_junos_223_r1]
...
```

NOTE: Use “local” keyword for operational commands in FIPS mode. For example, `show version local`, and `show system uptime local`.

Configuring Security Administrator and FIPS User Identification and Access

IN THIS SECTION

- Configuring Security Administrator Login Access | 32
- Configuring FIPS User Login Access | 33

Security Administrator and FIPS users perform all configuration tasks for Junos OS in FIPS mode and issue all Junos OS in FIPS mode statements and commands. Security Administrator and FIPS user configurations must follow Junos OS in FIPS mode guidelines.

Configuring 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** permission bits set is a Security Administrator. In most cases the **super-user** class suffices for the Security Administrator.

To configure login access for a Security Administrator:

1. Log in to the device with the root password if you have not already done so, and enter configuration mode:

```
root@host:fips> configure
Entering configuration mode
[edit]
root@host:fips#
```

2. Name the user “security-administrator” and assign the Security Administrator a user ID (for example, **6400**) and a class (for example, **super-user**). When you assign the class, you assign the permissions—for example, **secret**, **security**, **maintenance**, and **control**.

```
[edit]
root@host:fips# set system login user crypto-officer uid 6400 class super-user
```

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

```
[edit]
root@host:fips# set system login user crypto-officer class super-user authentication plain-
text-password
```

4. Optionally, display the configuration:

```
[edit]
root@host:fips# edit system
[edit system]
root@host:fips# show
```

```
login {
    user crypto-officer {
        uid 6400;
        authentication {
            encrypted-password "<cipher-text>"; ## SECRET-DATA
        }
        class super-user;
    }
}
```

5. If you are finished configuring the device, commit the configuration and exit:

```
[edit]
root@host:fips# commit
commit completeroot@host:fips# exit
root@host:fips> exit
```

Otherwise, go on to "Configuring FIPS User Login Access" on page 33.

Configuring FIPS User Login Access

A **fips-user** is defined as any FIPS user that does not have the **secret**, **security**, **maintenance**, and **control** permission bits set. As the Security Administrator, you set up FIPS users.

To configure login access for a FIPS user:

1. Log in to the device with your Security Administrator password if you have not already done so, and enter configuration mode:

```
crypto-officer@host:fips> configure
Entering configuration mode
[edit]
crypto-officer@host:fips#
```

2. Give the user a username, assign the FIPS user a user ID (for example, **6401**) and a class (for example , read-only). When you assign the class, you assign the permissions—for example, **clear**, **configure**, **network**, **resetview**, and **view-configuration**.

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

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

```
[edit]
crypto-officer@host:fips# set system login user fips-user1 class operator authentication
plain-text-password
```

4. Optionally, display the configuration:

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

5. If you are finished configuring the device, commit the configuration and exit:

```
[edit]
crypto-officer@host:fips# commit
crypto-officer@host:fips> exit
```

RELATED DOCUMENTATION

Understanding Roles and Services for Junos OS in Common Criteria and FIPS | 13

CHAPTER 3

Configuring Administrative Credentials and Privileges

IN THIS CHAPTER

- Understanding the Associated Password Rules for an Authorized Administrator | **37**
- Configuring a Network Device collaborative Protection Profile for an Authorized Administrator | **39**
- Customize Time | **40**
- Configuring Inactivity Timeout Period Configuration, and Local and Remote Idle Session Termination | **41**

Understanding the Associated Password Rules for an Authorized Administrator

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



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

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

- Easy to remember so that users are not tempted to write it down.
- Changed periodically.
- Private and not shared with anyone.
- Contain a minimum of 10 characters. The minimum password length is 10 characters.
- Include both alphanumeric and punctuation characters, composed of any combination of upper and lowercase letters, numbers, and special characters such as, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. There should be at least a change in one case, one or more digits, and one or more punctuation marks.
- Contain character sets. Valid character sets include uppercase letters, lowercase letters, numbers, punctuation, and other special characters.

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

- Contain the minimum number of character sets or character set changes. The minimum number of character sets required in plain-text passwords in Junos FIPS is 3.

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

- Contain the minimum number of characters required for a password. By default, Junos OS passwords must be at least 6 characters long. The valid range for this option is 10 to 20 characters.

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

NOTE: The device supports ECDSA (P-256, P-384, and P-521) and RSA (2048, 3072, and 4092 modulus bit length) key-types.

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

Weak passwords are:

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

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

NOTE: Passwords should be changed periodically.

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 9

Configuring a Network Device collaborative Protection Profile for an Authorized Administrator

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

An NDcPP Version 2.2e authorized administrator must have all permissions, including the ability to change the router configuration.

To configure an authorized administrator:

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

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

2. Configure the hashing algorithm used for password storage as sha512.

```
root@host# set system login password format sha512
```



NOTE: For your security devices, the default password algorithm is sha512, and it is not necessary to configure the plain-text passwords for EX and QFX series switches.

3. Commit the changes.

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

4. Define your NDcPP Version 2.2e authorized administrator.

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

5. Load an SSH key file that was previously generated using ssh-keygen. This command loads RSA (SSH version 2), or ECDSA (SSH version 2).

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

6. Set the log-key-changes configuration statement to log when SSH authentication keys are added or removed.

```
[edit]
root@host#set system services ssh log-key-changes
```

> **NOTE**: When the `log-key-changes` configuration statement is enabled and committed (with the `commit` command in configuration mode), Junos OS logs the changes to the set of authorized SSH keys for each user (including the keys that were added or removed). Junos OS logs the differences since the last time the `log-key-changes` configuration statement was enabled. If the `log-key-changes` configuration statement was never enabled, then Junos OS logs all the authorized SSH keys.

7. Commit the changes.

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

RELATED DOCUMENTATION

Understanding the Associated Password Rules for an Authorized Administrator | 37

Customize Time

To customize time, disable NTP and set the date.

1. Disable NTP.

```
[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. Setting date and time. Date and time format is YYYYMMDDHHMM.ss.

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

Configuring Inactivity Timeout Period Configuration, and Local and Remote Idle Session Termination

IN THIS SECTION

- Configure Session Termination | 41
- Sample Output for Local Administrative Session Termination | 43
- Sample Output for Remote Administrative Session Termination | 43
- Sample Output for User Initiated Termination | 44

Configure Session Termination

Terminate the session after the security administrator specifies inactive 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.

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

5. Define login class.

```
[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

```
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 22.3R1 Kernel 64-bit  JNPR-12.1-20220628.HEAD__ci_fbs
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

```
ssh NDcPPv2-user@host
Password:
Last login: Sun Jun 23 22:48:05 2019
--- JUNOS 22.3R1 Kernel 64-bit  JNPR-12.1-20220628.HEAD__ci_fbs
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 22.3R1 Kernel 64-bit  JNPR-12.1-20220628.HEAD__ci_fbs
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 Termination

```
ssh NDcPPv2-user@host
Password:
Last login: Sun Jun 23 22:48:05 2019
--- JUNOS 22.3R1 Kernel 64-bit  JNPR-12.1-20220628.HEAD__ci_fbs
NDcPPv2-user@host> exit

Connection to host closed.
```

CHAPTER 4

Configuring SSH and Console Connection

IN THIS CHAPTER

- Configuring a System Login Message and Announcement | **46**
- Configuring SSH on the Evaluated Configuration | **47**
- Limiting the Number of User Login Attempts for SSH Sessions | **49**

Configuring a System Login Message and Announcement

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

To configure a system login message through console or management interface, use the following command:

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

To configure system announcement, use the following command:

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

NOTE:

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

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration | 47

Configuring SSH on the Evaluated Configuration

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

- Before you begin, log in with your root account on the device.

To configure SSH on the device:

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

```
[edit ]
root@host# set system services ssh hostkey-algorithm-list rsa
root@host# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
root@host# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
root@host# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
```

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

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

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

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

4. Specify the ciphers allowed for protocol version 2.

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

NOTE: To disable SSH service, you can deactivate SSH configurations:

```
root@host# deactivate system services ssh
```

NOTE: To disable Netconf service, you can deactivate netconf configurations:

```
root@host# deactivate system services netconf ssh
```

Supported SSH hostkey algorithm:

```
ssh-ecdsa            Allow generation of ECDSA host-key
ssh-rsa              Allow generation of RSA host-key
```

Supported SSH key-exchange algorithm:

```
dh-group14-sha1      The RFC 4253 mandated group14 with SHA1 hash
ecdh-sha2-nistp256   The EC Diffie-Hellman on nistp256 with SHA2-256
ecdh-sha2-nistp384   The EC Diffie-Hellman on nistp384 with SHA2-384
ecdh-sha2-nistp521   The EC Diffie-Hellman on nistp521 with SHA2-512
```

Supported MAC algorithm:

```
hmac-sha1            Hash-based MAC using Secure Hash Algorithm (SHA1)
hmac-sha2-256        Hash-based MAC using Secure Hash Algorithm (SHA2)
hmac-sha2-512        Hash-based MAC using Secure Hash Algorithm (SHA2)
```

Supported SSH ciphers algorithm:

```
aes128-cbc           128-bit AES with Cipher Block Chaining
aes128-ctr           128-bit AES with Counter Mode

aes256-cbc           256-bit AES with Cipher Block Chaining
aes256-ctr           256-bit AES with Counter Mode
```

RELATED DOCUMENTATION

Limiting the Number of User Login Attempts for SSH Sessions | 49

Limiting the Number of User Login Attempts for SSH Sessions

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

You configure the amount of time the device gets locked after failed attempts. The amount of time in minutes before the user can attempt to log in to the device after being locked out due to the number of failed login attempts specified in the `tries-before-disconnect` statement. When a user fails to correctly login after the number of allowed attempts specified by the `tries-before-disconnect` statement, the user must wait the configured amount of minutes before attempting to log in to the device again. During this lockout-period the remote session user still have access to the TOE through the console as the root user.

The `lockout-period` must be greater than zero. The range at which you can configure the `lockout-period` is one through 43,200 minutes.

```
[edit system login]
user@host# set retry-options lockout-period number
```

You can configure the device to limit the number of attempts to enter a password while logging through SSH. Using the following command, the connection.

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

Here, `tries-before-disconnect` is the number of times a user can attempt to enter a password when logging in. The connection closes if a user fails to log in after the number specified. The range is from 2 through 10, and the default value is 3.

The local administrator access will be maintained even if the remote administration is made permanently or temporarily unavailable due to the multiple failed login attempts. The console login for local administration will be available to the users during the lockout period.

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

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

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

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

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

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

You can control user access through SSH. By configuring `ssh root-login deny` , you can ensure the root account remains active and continues to have local administrative privileges to the TOE even if other remote users are logged off.

```
[edit system]
user@host# set services ssh root-login deny
```

The SSH2 protocol provides secure terminal sessions utilizing the secure encryption. The SSH2 protocol enforces running the key-exchange phase and changing the encryption and integrity keys for the session. Key exchange is done periodically, after specified seconds or after specified bytes of data have passed over the connection. You can configure thresholds for SSH rekeying, FCS_SSHS_EXT.1.8 and FCS_SSHC_EXT.1.8. The TSF ensures that within the SSH connections the same session keys are used

for a threshold of no longer than one hour, and no more than one gigabyte of the transmitted data. When either of the thresholds are reached, a rekey must be performed.

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

Time limit before renegotiating session keys is 1 through 1440 minutes.

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

Data limit before renegotiating session keys is 51200 through 4294967295 byte.



NOTE: For SSH connection being unintentionally broken, we need to re-initiate the SSH connection to log in back to TOE.

RELATED DOCUMENTATION

Configuring SSH on the Evaluated Configuration | 47

5
CHAPTER

Configuring the Remote Syslog Server

IN THIS CHAPTER

- Syslog Server Configuration on a Linux System | **53**

Syslog Server Configuration on a Linux System

IN THIS SECTION

- Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server | 54

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

Use the configuration details and establish a session between the target of evaluation (TOE) and the audit server. Examine the traffic that passes between the audit server and the TOE during several activities, and the generated audit data to be transferred to the audit server.

Examine the TOE Summary Specification (TSS) to ensure that it specifies the means by which the audit data is transferred to the external audit server and how the trusted channel is provided.

The NDcPP logs capture 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

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

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

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

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

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

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

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

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

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

4. Set up NETCONF with SSH.

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

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

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

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

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

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

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

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

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

```
$ ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

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

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

10. The examples for syslog messages are listed below. Monitor the event log generated for admin actions on TOE as received on the syslog server. Examine the traffic that passes between the audit server and the TOE, observing that these data are not viewed during this transfer, and that they are successfully received by the audit server. Match the logs between local event and the remote event logged in a syslog server and record the particular software (such as name, version, and so on) used on the audit server during testing.

The following output shows test log results for syslog-server.

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

```
|                 |
|                 |
|               ..|
|        S       +|
|         .     Bo|
|          . . *.X|
|         . . o E@|
|          .   .BX|
+-----------------+
[host@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)
```

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 event logs generated on the TOE that are received on the syslog server.

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

Net configuration channel

```
host@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 syslogs and remote syslogs received were 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
..............
```

CHAPTER 6

Configuring Audit Log Options

IN THIS CHAPTER

- Configuring Audit Log Options in the Evaluated Configuration | **62**
- Sample Code Audits of Configuration Changes | **63**

Configuring Audit Log Options in the Evaluated Configuration

IN THIS SECTION

- Configuring Audit Log Options | 62

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

Configuring Audit Log Options

To configure audit log options for your devices:

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

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

2. Specify the file in which to log data.

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

3. Specify the size of files to be archived.

```
[edit system syslog]
root@host#set file  Audit_logs archive size 10m
```

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

```
[edit system syslog]
root@host#set file  Audit_logs explicit-priority
```

5. Log system messages in a structured format.

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

RELATED DOCUMENTATION

Sample Code Audits of Configuration Changes | 63

Sample Code Audits of Configuration Changes

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

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

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

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

```
    }
}
```

Example: System Logging of Configuration Changes

This example shows a sample configuration and makes changes to users and secret data.

```
[edit system]
location {
    country-code US;
    building B1;
}
...
login {
    message "UNAUTHORIZED USE OF THIS ROUTER\n\tIS STRICTLY PROHIBITED!";
        user admin {
            uid 2000;
             class super-user;
        authentication {
            encrypted-password “$ABC123”;
               # SECRET-DATA
        }
    }
    password {
        format sha512;
    }
}
radius-server 192.0.2.15 {
    secret “$ABC123” # SECRET-DATA
}
services {
    ssh;
}
syslog {
    user *{
        any emergency;
    }
    file messages {
        any notice;
        authorization info;
    }
    file interactive-commands {
```

```
        interactive-commands any;
    }
}
...
...
```

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

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

Table 3 on page 65 shows sample for syslog auditing for NDcPPv2.2e:

Table 3: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FAU_GEN.1	None	None	
FAU_GEN.2	None	None	
FAU_STG_EXT.1	None	None	
FAU_STG.1	None	None	

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event 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_COP.1(1)/ KeyedHashCMAC	None	None	
FCS_RBG_EXT.1	None	None	
FIA_PMG_EXT.1	None	None	

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address)	Successful Local Login Jan 3 09:59:36 login[7637]: LOGIN_INFORMATION: User root logged in from host [unknown] on device ttyu0 Jan 3 09:59:36 login[7637]: LOGIN_ROOT: User root logged in as root from host [unknown] on device ttyu0 Unsuccessful Local Login Jan 3 09:57:52 login[7637]: LOGIN_PAM_AUTHENTICATION_ERROR: Failed password for user root Jan 3 09:57:52 login[7637]: LOGIN_FAILED: Login failed for user root from host ttyu0 Successful Remote Login Jan 3 09:32:07 mgd[47035]: UI_AUTH_EVENT: Authenticated user 'test1' assigned to class 'j-read-only' Jan 3 09:32:07 mgd[47035]: UI_LOGIN_EVENT: User

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
			'test1' login, class 'j-read-only' [47035], ssh-connection '10.1.5.153 36784 10.1.2.68 22', client-mode 'cli' Unsuccessful Remote Login Jan 3 09:26:56 sshd: SSHD_LOGIN_FAILED: Login failed for user 'test1' from host '10.1.5.153'

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address)	Successful Local Login Jan 3 09:59:36 login[7637]: LOGIN_INFORMATION: User root logged in from host [unknown] on device ttyu0 Jan 3 09:59:36 login[7637]: LOGIN_ROOT: User root logged in as root from host [unknown] on device ttyu0 Unsuccessful Local Login Jan 3 09:57:52 login[7637]: LOGIN_PAM_ AUTHENTICATION_ER ROR: Failed password for user root Jan 3 09:57:52 login[7637]: LOGIN_FAILED: Login failed for user root from host ttyu0 Successful Remote Login Jan 3 09:32:07 mgd[47035]: UI_AUTH_EVENT: Authenticated user 'test1' assigned to class 'j-read-only' Jan 3 09:32:07 mgd[47035]: UI_LOGIN_EVENT: User 'test1' login, class 'j-read-only' [47035], ssh-

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
			connection '10.1.5.153 36784 10.1.2.68 22', client-mode 'cli' Unsuccessful Remote Login Jan 3 09:26:56 sshd: SSHD_LOGIN_FAILED: Login failed for user 'test1' from host '10.1.5.153'
FIA_UAU.7	None	None	
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None	`mgd[23878]: UI_CMDLINE_READ_LINE: User 'root', command 'request system software add /var/tmp/junos-install-ex-x86-64-22.3R1.tgz`
FMT_MTD.1/CoreData	None	None	
FMT_SMF.1	All management activities of TSF data	None	Refer to the audit events listed in this table.
FMT_SMR.2	None	None	
FPT_SKP_EXT.1	None	None	
FPT_APW_EXT.1	None	None	

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None	`mgd[23878]: UI_CMDLINE_READ_LINE: User 'root', command 'request system software add /var/tmp/junos-install-ex-x86-64-22.3R1.tgz`
FPT_STM_EXT.1 FTA_SSL_EXT.1 (if "terminate the session is selected)	The termination of a local interactive session by the session locking mechanism.	None	Jan 3 11:59:29 cli: UI_CLI_IDLE_TIMEOUT: Idle timeout for user 'root' exceeded and session terminated
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None	Jan 3 11:26:23 cli: UI_CLI_IDLE_TIMEOUT: Idle timeout for user 'root' exceeded and session terminated
FTA_SSL.4	The termination of an interactive session.	None	Local Jan 3 11:47:25 mgd[52521]: UI_LOGOUT_EVENT: User 'root' logout Remote Jan 3 11:43:33 sshd[52425]: Received disconnect from 10.1.5.153 port 36800:11: disconnected by user
FTA_TAB.1	None	None	

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.	Initiation of the trusted path Jan 3 12:09:00 sshd[53492]: Accepted keyboard-interactive/pam for root from 10.1.5.153 port 36802 ssh2 Termination of the trusted path Jan 3 12:09:03 sshd[53492]: Received disconnect from 10.1.5.153 port 36802:11: disconnected by user Jan 3 12:09:36 sshd: Failure of the trusted path SSHD_LOGIN_FAILED: Login failed for user 'root' from host '10.1.5.153'

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None	Initiation of the trusted path Jan 3 12:09:00 sshd[53492]: Accepted keyboard-interactive/pam for root from 10.1.5.153 port 36802 ssh2 Termination of the trusted path Jan 3 12:09:03 sshd[53492]: Received disconnect from 10.1.5.153 port 36802:11: disconnected by user Jan 3 12:09:36 sshd: Failure of the trusted path SSHD_LOGIN_FAILED: Login failed for user 'root' from host '10.1.5.153'
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, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, aes256-cbc

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store	Dec 28 22:20:23 veriexec[9371]: cannot validate /packages/db/pkginst.9286/manifest.ecerts: subject issuer mismatch: /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks/OU=Juniper CA/CN=PackageProductionTestEc_2017_NO_DEFECTS/emailAddress=ca@juniper.net
FIA_X509_EXT.2	None	None	
FPT_TUD_EXT.2	Failure of update	Reason for failure (including identifier of invalid certificate)	Dec 28 22:20:23 veriexec[9371]: cannot validate /packages/db/pkginst.9286/manifest.ecerts: subject issuer mismatch: /C=US/ST=CA/L=Sunnyvale/O=Juniper Networks/OU=Juniper CA/CN=PackageProductionTestEc_2017_NO_DEFECTS/emailAddress=ca@juniper.net
FMT_MOF.1/Functions	None	None	

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FMT_MOF.1/Services	None	None	
FMT_MTD.1/CryptoKeys	None	None	
FCS_MACSEC_EXT.1	Session establishment	Secure Channel Identifier (SCI)	Apr 10 20:43:35 dot1xd[6622]: DOT1XD_MKA_ SECURE_CHANNEL_ CREATED: Macsec receive secure channel created for 64:87:88:5a :19:30 on interface xe-0/0/0
FCS_MACSEC_EXT.1.7	Creation of Connectivity Association	Connectivity Association Key Names	Apr 10 20:43:38 dot1xd[6622]: DOT1XD_MKA_SECUR E_ ASSOCIATION_ESTABLI SHED: Macsec secure association established with an :2 on interface xe-0/0/0
FCS_MACSEC_EXT.3.1	Creation and update of Secure Association Key	Creation and update times	Apr 29 16:01:49 fpc0 vsc8584_macsec_rx _sa_create: ifd 148 (ge-0/0/0), port_no 0, vsc8584_handle 0x1543be98, an 3, key 0x18ccb058, lowest_pn 1, sci 0x18ccb044

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated

FIA_AFL.1	Administrator lockout due to excessive authentication failures	None	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:

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated

			`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:
			
			`login: security-administrator`
			
			`Password:`
			
			`Last login: Tue Jan 10 15:01:43 on ttyu0`
			
			`--- JUNOS 22.3R1-S1.3 Kernel 64-bit JNPR-12.1-20221021.ecb908b2_bui`
			
			`security-administrator@bm-a:fips>`
			
			`[edit]`

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
			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

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FPT_RPL.1	Detected replay attempt	None	Apr 15 10:05:16.142910 MKA actor #0 received duplicate or delayed PDU Apr 15 10:05:16.142932 MKA actor #0 received MKPDU, SCI 3C:94:D5:A0:A0:07/1, MI 27:D7:9F:97:53:CF:EF:86:00:52:C1:78, MN 1530

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).	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

NOTE: We are not claiming NTP as part of FPT_STM_EXT.1 SFR. However, in our configuration guide we have leveraged activate/deactivate NTP services to validate MACsec tolerance and MACsec key-chain.

Table 3: Auditable Events *(Continued)*

Requirement	Auditable Events	Additional Audit Record Contents	How event generated
FPT_TST_EXT.1	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.

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/).

RELATED DOCUMENTATION

Configuring Audit Log Options in the Evaluated Configuration | 62

7

CHAPTER

Configuring Event Logging

IN THIS CHAPTER

- Event Logging Overview | **83**
- Configuring Event Logging to a Local File | **84**
- Interpreting Event Messages | **84**
- Logging Changes to Secret Data | **86**
- Login and Logout Events Using SSH | **86**
- Logging of Audit Startup | **87**

Event Logging Overview

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

In addition, Junos OS can:

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

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

- Changes to secret key data in the configuration.
- Committed changes.
- Login/logout of users.
- System startup.
- Failure to establish an SSH session.
- Establishment/termination of an SSH session.
- Changes to the (system) time.
- Termination of a remote session by the session locking mechanism.
- Termination of an interactive session.
- Changes to modification or deletion of cryptographic keys.
- Password resets.

In addition, Juniper Networks recommends that logging also:

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

RELATED DOCUMENTATION

Interpreting Event Messages | 84

Configuring Event Logging to a Local File

You can configure storing of audit information to a local file with the `syslog` statement. This example stores logs in a file named **messages**:

```
[edit system]
syslog {
    file messages;
}
```

RELATED DOCUMENTATION

Event Logging Overview | 83

Interpreting Event Messages

The following output shows a sample event message.

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

Table 4 on page 85 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 4: Fields in Event Messages

Field	Description	Examples
timestamp	Time when the message was generated, in one of two representations: • *MMM-DD HH:MM:SS.MS+/-HH:MM,* is the month, day, hour, minute, second and millisecond in local time. The hour and minute that follows the plus sign (+) or minus sign (-) is the offset of the local time zone from Coordinated Universal Time (UTC). • *YYYY-MM-DDTHH:MM:SS.MSZ* is the year, month, day, hour, minute, second and millisecond in UTC.	`Jul 24 17:43:28` is the timestamp expressed as local time in the United States. `2012-07-24T09:17:15.719Z` is 9:17 AM UTC on 24 July 2012.
hostname	Name of the host that originally generated the message.	`router1`
process	Name of the Junos OS process that generated the message.	`mgd`
processID	UNIX process ID (PID) of the Junos OS process that generated the message.	`4153`
TAG	Junos OS system log message tag, which uniquely identifies the message.	`UI_DBASE_LOGOUT_EVENT`
username	Username of the user initiating the event.	“admin”
message-text	English-language description of the event .	`set: [system radius-server 1.2.3.4 secret]`

RELATED DOCUMENTATION

Event Logging Overview | 83

Logging Changes to Secret Data

The following are examples of audit logs of events that change the secret data. Whenever there is a change in the configuration example, the syslog event should capture the below logs:

```
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]
```

Everytime a configuration is updated or changed, the syslog should capture these logs:

```
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]
```

RELATED DOCUMENTATION

Interpreting Event Messages | 84

Login and Logout Events Using SSH

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

```
Dec 20 23:17:35  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:42  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:53  bilbo sshd[16645]: Accepted password for op from 172.17.58.45 port 1673 ssh2
```

```
Dec 20 23:17:53  bilbo mgd[16648]: UI_AUTH_EVENT: Authenticated user 'op' at permission level
                                   'j-operator'
Dec 20 23:17:53  bilbo mgd[16648]: UI_LOGIN_EVENT: User 'op' login, class 'j-operator' [16648]
Dec 20 23:17:56  bilbo mgd[16648]: UI_CMDLINE_READ_LINE: User 'op', command 'quit '
Dec 20 23:17:56  bilbo mgd[16648]: UI_LOGOUT_EVENT: User 'op' logout
```

RELATED DOCUMENTATION

Interpreting Event Messages | 84

Logging of Audit Startup

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

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

RELATED DOCUMENTATION

Login and Logout Events Using SSH | 86

8

CHAPTER

Configure MACsec on QFX5120-48YM

IN THIS CHAPTER

- Overview of Media Access Control Security (MACsec) in FIPS mode | **89**
- Configure MACsec | **91**

Overview of Media Access Control Security (MACsec) in FIPS mode

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

MACsec allows you to secure point to point Ethernet link for almost all traffic, including frames from the Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol (ARP), and other protocols that are not typically secured on an Ethernet link because of limitations with other security solutions. MACsec can be used in combination with other security protocols such as IP Security (IPsec) and Secure Sockets Layer (SSL) to provide end-to-end network security.

MACsec is standardized in IEEE 802.1AE. The IEEE 802.1AE standard can be seen on the IEEE organization website at IEEE 802.1: BRIDGING & MANAGEMENT.

Each implementation of an algorithm is checked by a series of known answer test (KAT) self-tests and crypto algorithms validations (CAV). The following cryptographic algorithms are added specifically for MACsec.

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

Pre-shared key configurations for both connectivity association key name (CKN) and connectivity association key (CAK):

```
crypto-officer@hostname:fips# prompt security macsec connectivity-association connectivity-
association-name pre-shared-key cak
New cak (secret):
Retype new cak (secret):
```

```
crypto-officer@hostname: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 define a user defined name for the *ca_name* variable option and a user defined connectivity association key name in hexadecimal format for *ckn* variable option.

A pre-shared key is exchanged between directly-connected links to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK. The CKN and CAK must match on both ends of a link to create a MACsec-secured link.

If you configure 128 bit cipher suite, then CAK can be 32 digit hexadecimal number. If you configure 256 bit cipher suite, then CAK can be 64 digit hexadecimal number.



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 64 digits of a CKN or all 32 digits of a CAK, the system auto-configures all the remaining digits to 0. However, you will receive a warning message when you commit the configuration.

After the successful exchange and verification of the pre-shared keys by both ends of the link, the MACsec Key Agreement (MKA) protocol enables and manages the secure link. The MKA protocol then elects one of the two directly-connected devices as the key server. The key server then shares a random security with the other device over the MACsec-secure point-to-point link. The key server continues to periodically create and share a random security key with the other device over the MACsec-secured point-to-point link as long as MACsec is enabled.

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

RELATED DOCUMENTATION

- Understanding Media Access Control Security (MACsec)

Configure MACsec

IN THIS SECTION

- Customizing Time | 91
- Configuring MACsec on a Device Running Junos OS | 92
- Configuring Static MACsec with Layer 3 Traffic | 93
- Configuring MACsec with keychain using Layer 3 Traffic | 97
- Configuring Static MACsec for Layer 2 Traffic | 103
- Configuring MACsec with keychain for Layer 2 Traffic | 108
- Disable and Restart MACsec Sessions | 115

We can configure MACsec to secure point-to-point Ethernet links connecting your device with MACsec-capable MICs. Each point-to-point Ethernet link that you want to secure using MACsec must be configured independently. We can enable MACsec on device-to-device links using static connectivity association key (CAK) security mode.

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

Customizing Time

To customize time, disable NTP and set the date.

1. Disable NTP.

```
[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. Setting date and time. Date and time format is YYYYMMDDHHMM.ss

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



NOTE: We are not claiming NTP as part of FPT_STM_EXT.1 SFR. However, in our configuration guide we have leveraged activate/deactivate NTP services to validate MACsec tolerance and MACsec key-chain.

Configuring MACsec on a Device Running Junos OS

To configure MACsec on a device running Junos OS:

1. Configure the MACsec security mode as for the connectivity association.

```
[edit]
security-administrator@hostname:fips#  set security macsec connectivity-association
connectivity-association-name exclude-protocol protocol-name
security-administrator@hostname:fips#  set security macsec connectivity-association
connectivity-association-name include-sci
security-administrator@hostname:fips#  set security macsec connectivity-association
connectivity-association-name mka key-server-priority priority-number
security-administrator@hostname:fips#  set security macsec connectivity-association
connectivity-association-name mka transmit-interval interval
security-administrator@hostname:fips#  set security macsec connectivity-association
connectivity-association-name offset offset-number
```



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

2. Create the pre-shared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

```
[edit]
security-administrator@hostname:fips# prompt security macsec connectivity-association
```

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



NOTE: 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 from `0` through `65535`.

3. Set the MKA to security mode.

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



NOTE: `CA1` is an example of *connectivity-association-name* configured.

4. Assign the configured connectivity association with a specified MACsec interface.

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

Configuring Static MACsec with Layer 3 Traffic

To configure Static MACsec using ICMP traffic between device R0 and device R1:

In R0:

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

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

2. Set the trace option values.

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

3. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1 security-mode static-cak
```

5. Set the MKA key server priority.

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

6. Set the MKA transmit interval.

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

7. Enable the MKA secure.

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

8. Assign the connectivity association to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
connectivity-association CA1
security-administrator@hostname: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 connectivity association key name (CKN) and connectivity association key (CAK)

```
[edit]
security-administrator@hostname:fips#  set security macsec connectivity-association CA1 pre-
shared-key ckn 2345678922334455667788992222333444555666777888999222233334444555 5
security-administrator@hostname:fips#  prompt security macsec connectivity-association CA1
pre-shared-key cak
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  set security macsec connectivity-association CA1
offset 30
```

2. Set the trace option values.

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

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

3. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

5. Set the MKA transmit interval.

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

6. Enable the MKA secure.

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

7. Assign the connectivity association to an interface.

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

Configuring MACsec with keychain using Layer 3 Traffic

Synchronize both macsec endpoint devices to NTP as both device's time should be the same for key start time triggers. To configure MACsec with keychain using ICMP traffic between device R0 and device R1:

In R0:

1. Assign a tolerance value to the authentication key chain.

```
[edit]
security-administrator@hostname: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 upto 64 keys. For example, you can refer the following 4 keys:

```
[edit]
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 start-time 2018-03-20.20:35
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 key-name 2345678922334455667788992223334445556667778889992222333344445552
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 start-time 2018-03-20.20:37
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 key-name 2345678922334455667788992223334445556667778889992222333344445553
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 start-time 2018-03-20.20:39
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 key-name 2345678922334455667788992223334445556667778889992222333344445554
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 start-time 2018-03-20.20:41
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@hostname: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 upto 64 secret keys. For example, you can refer the following 4 secret keys:

```
[edit]
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@hostname:fips# set security macsec connectivity-association CA1
offset 50
security-administrator@hostname: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@hostname:fips# set security macsec traceoptions file MACsec.log
security-administrator@hostname:fips# set security macsec traceoptions file size 4000000000
security-administrator@hostname:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

To configure MACsec with keychain for Layer 3 Traffic:

In R1:

1. Assign a tolerance value to the authentication key chain.

```
[edit]
security-administrator@hostname: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 upto 64 keys. For example, you can refer the following 4 keys:

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

```
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@hostname: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 upto 64 secret keys. For example, you can refer the following 4 secret keys:

```
[edit]
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@hostname:fips# set security macsec connectivity-association CA1
offset 50
security-administrator@hostname: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 10G/xe interfaces macsec configuration only.
   - You can use the XPN ciphers `AES-GCM-XPN-128` and `AES-GCM-XPN-256` for 40G and 100G rates macsec configuration. You can also use the XPN ciphers `AES-GCM-XPN-128` and `AES-GCM-XPN-256` for 10G/xe interfaces macsec configuration, if it supports.

5. Set the trace option values.

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

6. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

8. Set the MKA key server priority.

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

9. Set the MKA transmit interval.

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

10. Enable the MKA secure.

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

11. Assign the connectivity association to an interface.

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

Configuring Static MACsec for Layer 2 Traffic

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

In R0:

1. Set the MKA key server priority.

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

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.

```
[edit]
security-administrator@hostname: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 is *2345678922334455667788992222333412345678922334455667788992222333 41*.

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

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

4. Set the trace option values.

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

5. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

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

In R1:

1. 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.

   ```
   [edit]
   security-administrator@hostname: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 is 2345678922334455667788992223334123456789223344556677889922233341.

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

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

3. Set the trace option values.

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

4. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

6. Set the MKA key server priority.

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

7. Set the MKA transmit interval.

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

8. Enable the MKA secure.

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

9. Assign the connectivity association to an interface.

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

10. Configure VLAN tagging.

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

Configuring MACsec with keychain for Layer 2 Traffic

Synchronize both macsec endpoint devices to NTP as both device's time should be the same for key start time triggers. To configure MACsec with keychain for ICMP traffic between device R0 and device R1:

In R0:

1. Assign a tolerance value to the authentication key chain.

```
[edit]
security-administrator@hostname: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 upto 64 keys. For example, you can refer the following 4 keys:

```
[edit]
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 start-time 2018-03-20.20:35
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 key-name 2345678922334455667788992223334445556667778889992222333344445552
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 start-time 2018-03-20.20:37
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 key-name 2345678922334455667788992223334445556667778889992222333344445553
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 start-time 2018-03-20.20:39
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 key-name 2345678922334455667788992223334445556667778889992222333344445554
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 start-time 2018-03-20.20:41
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@hostname: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 upto 64 secret keys. For example, you can refer the following 4 secret keys:

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

```
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@hostname:fips# set security macsec connectivity-association CA1
cipher-suite gcm-aes-256
```

4. Set the trace option values.

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

5. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

In R1:

1. Assign a tolerance value to the authentication key chain.

```
[edit]
security-administrator@hostname: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 upto 64 keys. For example, you can refer the following 4 keys:

```
[edit]
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 key-name 2345678922334455667788992223333444555666777888999222233334444555 1
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 0 start-time 2018-03-20.20:35
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 key-name 2345678922334455667788992223333444555666777888999222233334444555 2
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 1 start-time 2018-03-20.20:37
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 key-name 2345678922334455667788992223333444555666777888999222233334444555 3
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 2 start-time 2018-03-20.20:39
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 key-name 2345678922334455667788992223333444555666777888999222233334444555 4
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 3 start-time 2018-03-20.20:41
security-administrator@hostname:fips#  set security authentication-key-chains key-chain
macsec-kc1 key 4 key-name 2345678922334455667788992223333444555666777888999222233334444555 5
security-administrator@hostname: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 upto 64 secret keys. For example, you can refer the following 4 secret keys:

```
[edit]
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname:fips#  prompt security authentication-key-chains key-chain
macsec-kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1 pre-
shared-key-chain macsec-kc1
security-administrator@hostname:fips# set security macsec connectivity-association CA1
cipher-suite gcm-aes-256
```

4. Set the trace option values.

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

5. Assign the trace to an interface.

```
[edit]
security-administrator@hostname:fips# set security macsec interfaces interface-name
traceoptions file mka_xe size 1g
security-administrator@hostname: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@hostname:fips# set security macsec connectivity-association CA1
security-mode static-cak
```

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

Disable and Restart MACsec Sessions

To disable and restart the MACsec sessions use the following configurations:

- 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
```

9

CHAPTER

Performing Self-Tests on a Device

IN THIS CHAPTER

- Understanding FIPS Self-Tests | **118**

Understanding FIPS Self-Tests

IN THIS SECTION

- Performing Power-On Self-Tests on the Device | 119

The cryptographic module enforces security rules to ensure that a device running the Juniper Networks Junos operating system (Junos OS) in FIPS mode of operation meets the security requirements of FIPS 140-3 Level 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 known answer test (KAT) self-tests:

- `kernel`—KAT for kernel cryptographic routines
- `MACSec`—KAT for MACsec cryptographic implementation
- `libmd`—KAT for libmd
- `OpenSSL v1.0.2`—KAT for OpenSSL v1.0.2 cryptographic implementation
- `OpenSSL`—KAT for OpenSSL cryptographic implementation
- `QuickSec 7.0`—KAT for Quicksec 7.0 Toolkit cryptographic implementation
- `QuickSec`—KAT for QuickSec Toolkit cryptographic implementation
- `SSH IPsec`—KAT for SSH IPsec Toolkit cryptographic implementation

The KAT self-tests are performed automatically at startup and reboot when FIPS mode of operation is enabled on the device. Conditional self-tests are also performed automatically to verify digitally signed software packages, generated random numbers, RSA and DSA key pairs, and manually entered keys.

On demand, self tests can be executed by entering `request system fips self-test` at command line.

If the KATs are completed successfully, the system log (syslog) file is updated to display the tests that were executed.

If the device fails a KAT, the device writes the details to a system log file, enters FIPS error state (panic), and reboots.

The `file show /var/log/messages` command displays the system log. See FPT_TST_EXT.1 under Table 3 on page 65.

Performing Power-On Self-Tests on the Device

Each time the cryptographic module is powered on, the module tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged.

The module displays the following status output while running the power-on self-tests:

```
@ 1556787428 mgd start
Creating initial configuration:  ...
mgd: Running FIPS Self-tests
mgd: Testing kernel KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   SHA-2-384 Known Answer Test:                     Passed
mgd:   SHA-2-512 Known Answer Test:                     Passed
mgd:   AES128-CMAC Known Answer Test:                   Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd: Testing MACSec KATS:
mgd:   AES128-CMAC Known Answer Test:                   Passed
mgd:   AES256-CMAC Known Answer Test:                   Passed
mgd:   AES-ECB Known Answer Test:                       Passed
mgd:   AES-KEYWRAP Known Answer Test:                   Passed
mgd:   KBKDF Known Answer Test:                         Passed
mgd: Testing libmd KATS:
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   SHA-2-512 Known Answer Test:                     Passed
mgd: Testing OpenSSL v1.0.2 KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   FIPS ECDSA Known Answer Test:                    Passed
mgd:   FIPS ECDH Known Answer Test:                     Passed
mgd:   FIPS RSA Known Answer Test:                      Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                 Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd:   AES-GCM Known Answer Test:                       Passed
```

```
mgd:   ECDSA-SIGN Known Answer Test:                    Passed
mgd:   KDF-IKE-V1 Known Answer Test:                    Passed
mgd:   KDF-SSH-SHA256 Known Answer Test:                Passed
mgd:   KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed
mgd:   KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed
mgd: Testing OpenSSL KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   FIPS ECDSA Known Answer Test:                    Passed
mgd:   FIPS ECDH Known Answer Test:                     Passed
mgd:   FIPS RSA Known Answer Test:                      Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                 Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd:   AES-GCM Known Answer Test:                       Passed
mgd:   ECDSA-SIGN Known Answer Test:                    Passed
mgd:   KDF-IKE-V1 Known Answer Test:                    Passed
mgd:   KDF-SSH-SHA256 Known Answer Test:                Passed
mgd:   KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed
mgd:   KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed
mgd: Testing QuickSec 7.0 KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                 Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                 Passed
mgd:   AES-CBC Known Answer Test:                       Passed
mgd:   AES-GCM Known Answer Test:                       Passed
mgd:   SSH-RSA-ENC Known Answer Test:                   Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                  Passed
mgd:   SSH-ECDSA-SIGN Known Answer Test:                Passed
mgd:   KDF-IKE-V1 Known Answer Test:                    Passed
mgd:   KDF-IKE-V2 Known Answer Test:                    Passed
mgd: Testing QuickSec KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
mgd:   HMAC-SHA2-224 Known Answer Test:                 Passed
```

```
mgd:   HMAC-SHA2-256 Known Answer Test:                Passed
mgd:   HMAC-SHA2-384 Known Answer Test:                Passed
mgd:   HMAC-SHA2-512 Known Answer Test:                Passed
mgd:   AES-CBC Known Answer Test:                      Passed
mgd:   AES-GCM Known Answer Test:                      Passed
mgd:   SSH-RSA-ENC Known Answer Test:                  Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                 Passed
mgd:   KDF-IKE-V1 Known Answer Test:                   Passed
mgd:   KDF-IKE-V2 Known Answer Test:                   Passed
mgd: Testing SSH IPsec KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:        Passed
mgd:   DES3-CBC Known Answer Test:                     Passed
mgd:   HMAC-SHA1 Known Answer Test:                    Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                Passed
mgd:   AES-CBC Known Answer Test:                      Passed
mgd:   SSH-RSA-ENC Known Answer Test:                  Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                 Passed
mgd:   KDF-IKE-V1 Known Answer Test:                   Passed
mgd: Testing file integrity:
mgd:   File integrity Known Answer Test:               Passed
mgd: Testing crypto integrity:
mgd:   Crypto integrity Known Answer Test:             Passed
```



NOTE: The module implements cryptographic libraries and algorithms that are not utilized in the approved mode of operation.

10
CHAPTER

Configuration Statements

IN THIS CHAPTER

- fips | 123
- level | 124

fips

IN THIS SECTION

- Syntax | 123
- Hierarchy Level | 123
- Description | 123
- Required Privilege Level | 124
- Release Information | 124

Syntax

```
fips {
   level level;
   }
```

Hierarchy Level

```
[edit system]
```

Description

Configure Junos OS Federal Information Processing Standard (FIPS) mode features on a device.

The remaining statements are explained separately.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.

level

IN THIS SECTION

- Syntax | 124
- Hierarchy Level | 125
- Description | 125
- Options | 125
- Required Privilege Level | 125
- Release Information | 125

Syntax

```
level level;
```

Hierarchy Level

```
[edit system fips]
```

Description

Set the level for the Junos OS Federal Information Processing Standards (FIPS) mode on the device. Setting the FIPS level to a value other than the default, 0 (zero), enables FIPS mode on the device.

Compared to non-FIPS mode, Junos OS in FIPS mode is a nonmodifiable operational environment with limitations.

Options

level FIPS level on a device, from level 1 (lowest) through level 4 (highest). At level 0 (the default), the device is in non-FIPS mode.

- Range: 0 through 4



NOTE: To enable FIPS mode on Junos on an EX or QFX series switch, set level to 1. Only level 1 is supported.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.

11

CHAPTER

Operational Commands

IN THIS CHAPTER

- request system zeroize | 127

request system zeroize

IN THIS SECTION

- Syntax | 127
- Description | 127
- Options | 127
- Required Privilege Level | 128
- Output Fields | 128
- Sample Output | 128
- Release Information | 128

Syntax

request system zeroize

Description

Erase and replace with zeros all user-created data from Routing Engines.

Options

none—Zeroize all Routing Engines in Junos OS in FIPS mode. You must verify the request by typing **yes** to proceed. This command is restricted to Crypto Officers because the `maintenance` permission bit is one of the permission bits, along with `secret` and `control`, that distinguishes Crypto Officers from other FIPS users.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request system zeroize

```
crypto-officer@switch:fips> request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files?. In case of Dual RE system, both Routing
Engines will be zeroized [yes,no] (no) yes

warning: ipsec-key-management subsystem not running - not needed by configuration.
warning: zeroizing localre
```

Release Information

Command introduced in Junos OS Release 12.1.