JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

FIPS Evaluated Configuration Guide for MX960, MX480, and MX240 Devices

Published
2023-05-11

RELEASE
20.3X75-D30

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 FIPS Evaluated Configuration Guide for MX960, MX480, and MX240 Devices
20.3X75-D30
Copyright © 2023 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

1 Overview

Understanding Junos OS in FIPS Mode | 2

Understanding FIPS Terminology and Supported Cryptographic Algorithms | 4

Identifying Secure Product Delivery | 7

Understanding Management Interfaces | 8

2 Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | 11

3 Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS | 14

Understanding the Operational Environment for Junos OS in FIPS Mode | 16

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 20

Downloading Software Packages from Juniper Networks | 22

Installing Software on a Device with Single Routing Engine | 22

Understanding Zeroization to Clear System Data for FIPS Mode | 25

Zeroizing the System | 27

Enabling FIPS Mode | 28

Configuring Crypto Officer and FIPS User Identification and Access | 30

- Configuring Crypto Officer Access | 30
- Configuring FIPS User Login Access | 32

4 Configuring SSH and Console Connection

Configuring SSH on the Evaluated Configuration for FIPS | 35

5 Configuring MACsec

Understanding Media Access Control Security (MACsec) in FIPS mode | 38

Configuring MACsec | 39

Customizing Time | **39**

Configuring MACsec on a Device Running Junos OS | **40**

Configuring Static MACsec with ICMP Traffic | **41**

Configuring MACsec with keychain using ICMP Traffic | **45**

Configuring Static MACsec for Layer 2 Traffic | **52**

Configuring MACsec with keychain for Layer 2 Traffic | **57**

6 **Configuring Event Logging**

Event Logging Overview | 67

Configuring Event Logging to a Local File | 68

Interpreting Event Messages | 68

Logging Changes to Secret Data | 70

Login and Logout Events Using SSH | 70

Logging of Audit Startup | 71

7 **Performing Self-Tests on a Device**

Understanding FIPS Self-Tests | 73

Example: Configure FIPS Self-Tests | 73

Hardware and Software Requirements | **74**

Overview | **74**

Configuration | **75**

Verification | **76**

Verifying the FIPS Self-Test | **76**

8 **Operational Commands**

request system zeroize | 82

request vmhost zeroize no-forwarding | 83

About This Guide

Use this guide to operate MX960, MX480, and MX240 devices in Federal Information Processing Standards (FIPS) 140-2 Level 1 environment. FIPS 140-2 defines security levels for hardware and software that perform cryptographic functions.

RELATED DOCUMENTATION

- Common Criteria and FIPS Certifications

CHAPTER 1

Overview

Understanding Junos OS in FIPS Mode | 2

Understanding FIPS Terminology and Supported Cryptographic Algorithms | 4

Identifying Secure Product Delivery | 7

Understanding Management Interfaces | 8

Understanding Junos OS in FIPS Mode

IN THIS SECTION

- Supported Platforms and Hardwares | 2
- About the Cryptographic Boundary on Your Device | 3
- How FIPS Mode Differs from Non-FIPS Mode | 3
- Validated Version of Junos OS in FIPS Mode | 3

Federal Information Processing Standards (FIPS) 140-2 defines security levels for hardware and software that perform cryptographic functions. This Juniper Networks router running the Juniper Networks Junos operating system (Junos OS) in *FIPS mode* comply with the FIPS 140-2 Level 1 standard.

Operating this router in a FIPS 140-2 Level 1 environment requires enabling and configuring FIPS mode on the devices from the Junos OS command-line interface (CLI).

The Crypto Officer enables FIPS mode in Junos OS and sets up keys and passwords for the system and other *FIPS users*.

Supported Platforms and Hardwares

For the features described in this document, the following platforms are used to qualify FIPS certification:

- MX960, MX480, and MX240 devices installed with RE-S-1800X4 and LC MPC7E-10G (https://www.juniper.net/us/en/products/routers/mx-series/mx960-universal-routing-platform.html, https://www.juniper.net/us/en/products/routers/mx-series/mx480-universal-routing-platform.html, and https://www.juniper.net/us/en/products/routers/mx-series/mx240-universal-routing-platform.html).
- MX960, MX480, and MX240 devices installed with RE-S-X6 and LC MPC7E-10G (https://www.juniper.net/us/en/products/routers/mx-series/mx960-universal-routing-platform.html, https://www.juniper.net/us/en/products/routers/mx-series/mx480-universal-routing-platform.html, and https://www.juniper.net/us/en/products/routers/mx-series/mx240-universal-routing-platform.html).

About the Cryptographic Boundary on Your Device

FIPS 140-2 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

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 MD5 are disabled.
- Weak or unencrypted management connections must not be configured.
- Passwords must be encrypted with strong one-way algorithms that do not permit decryption.
- Administrator passwords must be at least 10 characters long.

Validated Version of Junos OS in FIPS Mode

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

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 7

Understanding FIPS Terminology and Supported Cryptographic Algorithms

IN THIS SECTION

- Terminology | 4
- Supported Cryptographic Algorithms | 5

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

Terminology

Critical security parameter (CSP)	Security-related information—for example, secret and private cryptographic keys and authentication data such as passwords and personal identification numbers (PINs)—whose disclosure or modification can compromise the security of a cryptographic module or the information it protects. For details, see "Understanding the Operational Environment for Junos OS in FIPS Mode" on page 16.
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.
FIPS	Federal Information Processing Standards. FIPS 140-2 specifies requirements for security and cryptographic modules. Junos OS in FIPS mode complies with FIPS 140-2 Level 1.
FIPS maintenance role	The role the Crypto Officer assumes to perform physical maintenance or logical maintenance services such as hardware or software diagnostics. For FIPS 140-2 compliance, the Crypto Officer 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.

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 73.

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 devices for non-FIPS operation. The Crypto Officer can zeroize the system with a CLI operational command.

Supported Cryptographic Algorithms

Table 1 on page 6 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	Host (module): • ECDSA P-256 • SSH-RSA Client (user): • ECDSA P-256 • ECDSA P-384 • ECDSA P-521 • SSH-RSA	• AES CTR 128 • AES CTR 192 • AES CTR 256 • AES CBC 128 • AES CBC 256	• HMAC-SHA-1 • HMAC-SHA-256 • HMAC-SHA-512

Table 2 on page 6 lists the MACsec LC supported ciphers.

Table 2: MACsec LC Supported Ciphers

MACsec LC Supported Ciphers
AES-GCM-128
AES-GCM-256

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

BEST PRACTICE: For FIPS 140-2 compliance, use only FIPS-approved cryptographic algorithms In Junos OS in FIPS mode.

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, 192, or 256 bits to encrypt and decrypt data in blocks of 128 bits.

ECDH	Elliptic Curve Diffie-Hellman. A variant of the Diffie-Hellman key exchange algorithm that uses cryptography based on the algebraic structure of elliptic curves over finite fields. ECDH allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. The shared secret can be used either as a key or to derive another key for encrypting subsequent communications using a symmetric key cipher.
ECDSA	Elliptic Curve Digital Signature Algorithm. A variant of the Digital Signature Algorithm (DSA) that uses cryptography based on the algebraic structure of elliptic curves over finite fields. The bit size of the elliptic curve determines the difficulty of decrypting the key. The public key believed to be needed for ECDSA is about twice the size of the security level, in bits. ECDSA using the P-256, P-384, and P-521 curves can be configured under OpenSSH.
HMAC	Defined as “Keyed-Hashing for Message Authentication” in RFC 2104, HMAC combines hashing algorithms with cryptographic keys for message authentication. For Junos OS in FIPS mode, HMAC uses the iterated cryptographic hash functions SHA-1, SHA-256, and SHA-512 along with a secret key.
SHA-256 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, and SHA-512 produces a 512-bit hash digest.

RELATED DOCUMENTATION

Understanding FIPS Self-Tests | 73

Understanding Zeroization to Clear System Data for FIPS Mode | 25

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.

Understanding Management Interfaces

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

- Local Management Interfaces—The RJ-45 console port on the 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.

2
CHAPTER

Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | 11

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: Do not use control characters in passwords.

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

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

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

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

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

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

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

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

```
[ edit ]
administrator@host# set system login password format sha512
```

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

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.

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 7

CHAPTER 3

Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS | 14

Understanding the Operational Environment for Junos OS in FIPS Mode | 16

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 20

Downloading Software Packages from Juniper Networks | 22

Installing Software on a Device with Single Routing Engine | 22

Understanding Zeroization to Clear System Data for FIPS Mode | 25

Zeroizing the System | 27

Enabling FIPS Mode | 28

Configuring Crypto Officer and FIPS User Identification and Access | 30

Understanding Roles and Services for Junos OS

IN THIS SECTION

- Crypto Officer Role and Responsibilities | 15
- FIPS User Role and Responsibilities | 15
- What Is Expected of All FIPS Users | 16

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 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, 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. In contrast, the FIPS 140-2 standard defines two user roles: Crypto Officer and FIPS user. These roles are defined in terms of Junos OS user capabilities.

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

Crypto Officer performs all FIPS-mode-related configuration tasks and issue all statements and commands for Junos OS in FIPS mode. Crypto Officer and FIPS user configurations must follow the guidelines for Junos OS in FIPS mode.

Crypto Officer Role and Responsibilities

The Crypto Officer is the person responsible for enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode on a device. The Crypto Officer 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 Crypto Officer administer the system in a secure manner by keeping passwords secure and checking audit files.

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

NOTE: Junos OS in FIPS mode does not support the FIPS 140-2 maintenance role, which is different from the Junos OS maintenance permission.

Among the tasks related to Junos OS in FIPS mode, the Crypto Officer 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 Crypto Officer, can view the configuration. Only the user assigned as the Crypto Officer can modify the configuration.

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

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

What Is Expected of All FIPS Users

All FIPS users, including the Crypto Officer, 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 140-2 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 | 27

Understanding the Operational Environment for Junos OS in FIPS Mode

IN THIS SECTION

- Hardware Environment for Junos OS in FIPS Mode | 17
- Software Environment for Junos OS in FIPS Mode | 17
- Critical Security Parameters | 18

A Juniper Networks device running the Juniper Networks 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 140-2 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 which includes LC MPC7E-10G card. Each component forms a separate cryptographic module. Communications involving CSPs between these secure environments must take place using encryption.

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 nonmodifiable operational environment. To achieve this environment on the device, the system prevents the execution of any binary file that was not part of the certified Junos OS in FIPS mode distribution. When a device is in FIPS mode, it can run only Junos OS.

The Junos OS in FIPS mode software environment is established after the Crypto Officer 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 140-2 compliance, we recommend that you delete all user-created files and data by *zeroizing* the device before enabling FIPS mode.

Operating your device at FIPS Level 1 requires the use of tamper-evident labels to seal the Routing Engines into the chassis.

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 the Crypto Officer completes 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 deviceor Routing Engine as a cryptographic module.

Table 3 on page 19 lists CSPs on devices running Junos OS.

Table 3: Critical Security Parameters

CSP	Description	Zeroize	Use
SSHv2 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.
SSHv2 session keys	Session key used with SSHv2 and as a Diffie-Hellman private key. Encryption: AES-128, AES-192, AES-256. MACs: HMAC-SHA-1, HMAC-SHA-2-256, HMAC-SHA2-512. Key exchange: dh-group14-sha1, ECDH-sha2-nistp-256, ECDH-sha2-nistp-384, and ECDH-sha2-nistp-521.	Power cycle and terminate session.	Symmetric key used to encrypt data between host and client.
User authentication key	Hash of the user's password: SHA256, SHA512.	Zeroize command.	Used to authenticate a user to the cryptographic module.
Crypto Officer authentication key	Hash of the Crypto Officer's password: SHA256, SHA512.	Zeroize command.	Used to authenticate the Crypto Officer 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.

Table 3: Critical Security Parameters *(Continued)*

CSP	Description	Zeroize	Use
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.
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.

BEST PRACTICE: For FIPS compliance, configure the device over SSH connections because they are encrypted connections.

Local passwords are hashed with the SHA256 or SHA512 algorithm. 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.

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode

All passwords established for users by the Crypto Officer 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. Passwords must contain between 10 and 20 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 (`r00t`) 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.

Downloading Software Packages from Juniper Networks

You can download the Junos OS software package for your device 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/.

To download software packages from Juniper Networks:

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

RELATED DOCUMENTATION

> Installation and Upgrade Guide

Installing Software on a Device with Single Routing Engine

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

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

1. Download the software package as described in Downloading Software Packages from Juniper Networks.
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 Software 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:

   For REMX2K-X8:

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

   For RE1800:

   ```
   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 variable option *package* with the software package name.
     - **ftp://*hostname*/*pathname*/*package*.tgz**
     - **http://*hostname*/*pathname*/*package*.tgz**

6. Reboot the device to load the installation:

   For REMX2K-X8:

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

   For RE1800:

   ```
   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> show version
   Model: mx960
   Junos: 20.3X75-D30.1
   JUNOS OS Kernel 64-bit  [20210722.b0da34e0_builder_stable_11-204ab]
   JUNOS OS libs [20210722.b0da34e0_builder_stable_11-204ab]
   JUNOS OS runtime [20210722.b0da34e0_builder_stable_11-204ab]
   JUNOS OS time zone information [20210722.b0da34e0_builder_stable_11-204ab]
   ```

```
JUNOS network stack and utilities [20210812.200100_builder_junos_203_x75_d30]
JUNOS libs [20210812.200100_builder_junos_203_x75_d30]
JUNOS OS libs compat32 [20210722.b0da34e0_builder_stable_11-204ab]
JUNOS OS 32-bit compatibility [20210722.b0da34e0_builder_stable_11-204ab]
JUNOS libs compat32 [20210812.200100_builder_junos_203_x75_d30]
JUNOS runtime [20210812.200100_builder_junos_203_x75_d30]
JUNOS sflow mx [20210812.200100_builder_junos_203_x75_d30]
JUNOS py extensions2 [20210812.200100_builder_junos_203_x75_d30]
JUNOS py extensions [20210812.200100_builder_junos_203_x75_d30]
JUNOS py base2 [20210812.200100_builder_junos_203_x75_d30]
JUNOS py base [20210812.200100_builder_junos_203_x75_d30]
JUNOS OS crypto [20210722.b0da34e0_builder_stable_11-204ab]
JUNOS OS boot-ve files [20210722.b0da34e0_builder_stable_11-204ab]
JUNOS na telemetry [20.3X75-D30.1]
JUNOS Security Intelligence [20210812.200100_builder_junos_203_x75_d30]
JUNOS mx libs compat32 [20210812.200100_builder_junos_203_x75_d30]
JUNOS mx runtime [20210812.200100_builder_junos_203_x75_d30]
JUNOS RPD Telemetry Application [20.3X75-D30.1]
Redis [20210812.200100_builder_junos_203_x75_d30]
JUNOS probe utility [20210812.200100_builder_junos_203_x75_d30]
JUNOS common platform support [20210812.200100_builder_junos_203_x75_d30]
JUNOS Openconfig [20.3X75-D30.1]
JUNOS mtx network modules [20210812.200100_builder_junos_203_x75_d30]
JUNOS modules [20210812.200100_builder_junos_203_x75_d30]
JUNOS mx modules [20210812.200100_builder_junos_203_x75_d30]
JUNOS mx libs [20210812.200100_builder_junos_203_x75_d30]
JUNOS SQL Sync Daemon [20210812.200100_builder_junos_203_x75_d30]
JUNOS mtx Data Plane Crypto Support [20210812.200100_builder_junos_203_x75_d30]
JUNOS daemons [20210812.200100_builder_junos_203_x75_d30]
JUNOS mx daemons [20210812.200100_builder_junos_203_x75_d30]
JUNOS appidd-mx application-identification daemon [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services URL Filter package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services TLB Service PIC package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Telemetry [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services TCP-LOG [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services SSL [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services SOFTWIRE [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Stateful Firewall [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services RTCOM [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services RPM [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services PCEF package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services NAT [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Mobile Subscriber Service Container package
```

```
[20210812.200100_builder_junos_203_x75_d30]
JUNOS Services MobileNext Software package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Logging Report Framework package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services LL-PDF Container package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Jflow Container package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Deep Packet Inspection package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services IPSec [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services IDS [20210812.200100_builder_junos_203_x75_d30]
JUNOS IDP Services [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services HTTP Content Management package [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Crypto [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Captive Portal and Content Delivery Container package
[20210812.200100_builder_junos_203_x75_d30]
JUNOS Services COS [20210812.200100_builder_junos_203_x75_d30]
JUNOS AppId Services [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services Application Level Gateways [20210812.200100_builder_junos_203_x75_d30]
JUNOS Services AACL Container package [20210812.200100_builder_junos_203_x75_d30]
JUNOS SDN Software Suite [20210812.200100_builder_junos_203_x75_d30]
JUNOS Extension Toolkit [20210812.200100_builder_junos_203_x75_d30]
JUNOS Packet Forwarding Engine Support (wrlinux9) [20210812.200100_builder_junos_203_x75_d30]
JUNOS Packet Forwarding Engine Support (ulc) [20210812.200100_builder_junos_203_x75_d30]
JUNOS Packet Forwarding Engine Support (MXSPC3) [20.3X75-D30.1]
JUNOS Packet Forwarding Engine Support (X2000) [20210812.200100_builder_junos_203_x75_d30]
JUNOS Packet Forwarding Engine FIPS Support [20.3X75-D30.1]
JUNOS Packet Forwarding Engine Support (M/T Common)
[20210812.200100_builder_junos_203_x75_d30]
JUNOS Packet Forwarding Engine Support (aft)
```

Understanding Zeroization to Clear System Data for FIPS Mode

IN THIS SECTION

- Why Zeroize? | **26**
- When to Zeroize? | **26**

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.

Crypto Officer initiates the zeroization process by entering the operational command `request vmhost zeroize no-forwarding` for REMX2K-X8 and `request system zeroize` for RE1800.

> ⚠ **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 140-2 compliance, you must zeroize the system to remove sensitive information before disabling FIPS mode on the device.

When to Zeroize?

As Crypto Officer, 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 disabling FIPS mode of operation:** To begin repurposing your device for non-FIPS operation, perform zeroization before disabling FIPS mode 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.

Zeroizing the System

To zeroize your device, follow the below procedure:

1. Login to the device as Crypto Officer and from CLI, enter the following command.

   For REMX2K-X8:

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

   re0:
   ```

   For REMX2K-X8:

   ```
   crypto-officer@host> request system zeroize
   System Zeroization : Erase all data, including configuration and log files ?
   [yes,no] (no) yes

   re0:
   ```

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

   ```
   Erase all data, including configuration and log files?   [yes, no] (no) yes
   Erase all data, including configuration and log files? [yes, no] (no)
   yes
   re0:
   --------------------------------------------------------------------------
   warning: zeroizing re0
   ...
   ...
   ```

   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.

Enabling FIPS Mode

When Junos OS is installed on a device and the device is powered on, it is ready to be configured. Initially, you log in as the user root with no password. When you log in as root, your SSH connection is enabled by default.

As Crypto Officer, you must establish a root password conforming to the FIPS password requirements in "Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 20. When you enable FIPS mode in Junos OS on the device, you cannot configure passwords unless they meet this standard.

Local passwords are encrypted with the secure hash algorithm SHA256 or SHA512. 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.

To enable FIPS mode in Junos OS on the device:

1. Zeroize the device to delete all CSPs before entering FIPS mode. Refer to "Understanding Zeroization to Clear System Data for FIPS Mode" on page 25 section for details.
2. After the device comes up in 'Amnesiac mode', login using username `root` and password `""` (blank).

```
FreeBSD/amd64 (Amnesiac) (ttyu0)
login: root
--- JUNOS 20.3X75-D30.1 Kernel 64-bit JNPR-11.0-20190701.269d466_buil
root@:~ #  cli
root>
```

3. Configure root authentication with password at least 10 characters or more.

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

4. Load configuration onto device and commit new configuration. Configure crypto-officer and login with crypto-officer credentials.

5. Install `fips-mode` package needed for Routing Engine KATS.

```
root@hostname> request system software add optional://fips-mode.tgz
Verified fips-mode signed by PackageDevelopmentEc_2017 method ECDSA256+SHA256
```

6. For MX Series devices,
   - Configure chassis boundary fips by setting `set system fips chassis level 1` and `commit`.
   - Configure RE boundary fips by setting `set systems fips level 1` and commit.

   Device might display the **Encrypted-password must be re-configured to use FIPS compliant hash** warning to delete the older CSPs in the loaded configuration.
7. After deleting and reconfiguring CSPs, commit will go through and device needs reboot to enter FIPS mode.

```
[edit]
crypto-officer@hostname# commit
Generating RSA key /etc/ssh/fips_ssh_host_key
Generating RSA2 key /etc/ssh/fips_ssh_host_rsa_key
Generating ECDSA key /etc/ssh/fips_ssh_host_ecdsa_key
[edit]
system
reboot is required to transition to FIPS level 1
commit complete
[edit]
crypto-officer@hostname# run request vmhost reboot
```

8. After rebooting the device, FIPS self-tests will run and device enters FIPS mode.

```
crypto-officer@hostname:fips>
```

RELATED DOCUMENTATION

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 20

Configuring Crypto Officer and FIPS User Identification and Access

IN THIS SECTION

- Configuring Crypto Officer Access | 30
- Configuring FIPS User Login Access | 32

Crypto Officer enables FIPS mode on your device and performs all configuration tasks for Junos OS in FIPS mode and issue all Junos OS in FIPS mode statements and commands. Crypto Officer and FIPS user configurations must follow Junos OS in FIPS mode guidelines.

Configuring Crypto Officer Access

Junos OS in FIPS mode offers a finer granularity of user permissions than those mandated by FIPS 140-2.

For FIPS 140-2 compliance, any FIPS user with the `secret`, `security`, `maintenance`, and `control` permission bits set is a Crypto Officer. In most cases the `super-user` class suffices for the Crypto Officer.

To configure login access for a Crypto Officer:

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

```
root@hostname> edit
Entering configuration mode
[edit]
root@hostname#
```

2. Name the user `crypto-officer` and assign the Crypto Officer a user ID (for example, `6400`, which must be a unique number associated with the login account in the range of 100 through 64000) and a class (for example, `super-user`). When you assign the class, you assign the permissions—for example, `secret`, `security`, `maintenance`, and `control`.

For a list of permissions, see Understanding Junos OS Access Privilege Levels.

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

For example:

```
[edit]
root@hostname# 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 20, assign the Crypto Officer 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@hostname#  set system login user username class class-name authentication (plain-test-
password | encrypted-password)
```

For example:

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

4. Optionally, display the configuration:

```
[edit]
root@hostname#edit system
[edit system]
root@hostname#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@hostname# commit
commit complete
root@hostname# exit
```

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 Crypto Officer you set up FIPS users. FIPS users cannot be granted permissions normally reserved for the Crypto Officer—for example, permission to zeroize the system.

To configure login access for a FIPS user:

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

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

2. Give the user, a username, and assign the user a user ID (for example, `6401`, which must be a unique number in the range of 1 through 64000) and a class. When you assign the class, you assign the permissions—for example, `clear`, `network`, `resetview`, and `view-configuration`.

```
[edit]
crypto-officer@hostname:fips# set system login user username uid value class class-name
```

For example:

```
[edit]
crypto-officer@hostname: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 20, assign the FIPS user a plain-text password for login authentication. Set the password by typing a password after the prompts `New password` and `Retype new password`.

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

For example:

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

4. Optionally, display the configuration:

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

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

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

RELATED DOCUMENTATION

Understanding Roles and Services for Junos OS | 14

4

CHAPTER

Configuring SSH and Console Connection

Configuring SSH on the Evaluated Configuration for FIPS | 35

Configuring SSH on the Evaluated Configuration for FIPS

SSH through remote management interface allowed in the evaluated configuration. This topic describes how to configure SSH through remote management.

The following algorithms that needs to be configured to validate SSH for FIPS.

To configure SSH on the DUT:

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

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

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

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

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

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

4. Specify the ciphers allowed for protocol version 2.

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

```
user@host# set system services ssh ciphers aes256-ctr
user@host# set system services ssh ciphers aes192-cbc
user@host# set system services ssh ciphers aes192-ctr
```

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:

```
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
aes192-cbc           192-bit AES with Cipher Block Chaining
aes192-ctr           192-bit AES with Counter Mode
aes256-cbc           256-bit AES with Cipher Block Chaining
aes256-ctr           256-bit AES with Counter Mode
```

5
CHAPTER

Configuring MACsec

Understanding Media Access Control Security (MACsec) in FIPS mode | 38

Configuring MACsec | 39

Understanding 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

For MACsec, in configuration mode, use the `prompt` command to enter a secret key value of 64 hexadecimal characters for authentication.

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

RELATED DOCUMENTATION

- Understanding Media Access Control Security (MACsec)

Configuring MACsec

IN THIS SECTION

- Customizing Time | 39
- Configuring MACsec on a Device Running Junos OS | 40
- Configuring Static MACsec with ICMP Traffic | 41
- Configuring MACsec with keychain using ICMP Traffic | 45
- Configuring Static MACsec for Layer 2 Traffic | 52
- Configuring MACsec with keychain for Layer 2 Traffic | 57

We can configure MACsec to secure point-to-point Ethernet links connecting your device with MACsec-capable MICs, or on Ethernet links connecting your device to a host device such as a PC, phone, or server. 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 must download a MACsec feature license to use MACsec feature on your device. To purchase a feature license for MACsec, contact your Juniper Networks sales representative (https://www.juniper.net/us/en/how-to-buy/form.html). The Juniper sales representative will provide you with a feature license file and a license key. To add new license, see .

Customizing Time

To customize time, disable NTP and set the date.

1. Disable NTP.

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

2. Setting date and time. Date and time format is YYYYMMDDHHMM.ss

```
[edit]
crypto-officer@hostname:fips# set date 201803202034.00
crypto-officer@hostname:fips# set cli timestamp
```

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]
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name exclude-protocol protocol-name
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name include-sci
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name mka must-secure
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name mka key-server-priority priority-number
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name mka transmit-interval interval
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name no-encryption
crypto-officer@hostname:fips#  set security macsec connectivity-association connectivity-
association-name offset (0|30|50)
```

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

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association connectivity-
association-name pre-shared-key cak hexadecimal-number
crypto-officer@hostname:fips# set security macsec connectivity-association connectivity-
association-name pre-shared-key ckn hexadecimal-number
crypto-officer@hostname:fips# set security macsec connectivity-association connectivity-
association-name replay-protect{ replay-window-size number-of-packets
```

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

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

4. Set the MKA to security mode.

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association connectivity-
association-name security-mode security-mode
```

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

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

Configuring Static MACsec with ICMP 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]
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 pre-shared-key ckn 2345678922334455667788992223333444555666777888999222233334444555 5
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 pre-shared-key cak 23456789223344556677889922233344
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 offset 30
```

2. Set the trace option values.

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

3. Assign the trace to an interface.

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

5. Set the MKA key server priority.

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

6. Set the MKA transmit interval.

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

7. Enable the MKA secure.

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 mka should-
secure
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 include-sci
```

8. Assign the connectivity association to an interface.

```
[edit]
crypto-officer@hostname:fips# set security macsec interfaces interface-name connectivity-
association CA1
crypto-officer@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]
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 pre-shared-
key ckn 2345678922334455667788992223334445556667778889992222333344445555
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 pre-shared-
key cak 234567892233445566778899222333 44
crypto-officer@hostname:fips#  set security macsec connectivity-association CA1 offset 30
```

2. Set the trace option values.

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

3. Assign the trace to an interface.

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

5. Set the MKA transmit interval.

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

6. Enable the MKA secure.

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 mka should-
secure
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 include-sci
```

7. Assign the connectivity association to an interface.

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

Configuring MACsec with keychain using ICMP Traffic

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]
crypto-officer@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.

```
[edit]
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 start-time 2018-03-20.20:35
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 key-name 2345678922334455667788992223334445556667778889992222333344445552
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 start-time 2018-03-20.20:37
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 key-name 2345678922334455667788992223334445556667778889992222333344445553
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 start-time 2018-03-20.20:39
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 key-name 2345678922334455667788992223334445556667778889992222333344445554
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 start-time 2018-03-20.20:41
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 key-name 2345678922334455667788992223334445556667778889992222333344445555
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 start-time 2018-03-20.20:43
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 key-name 2345678922334455667788992223334445556667778889992222333344445556
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 start-time 2018-03-20.20:45
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
```

```
key 6 key-name 2345678922334455667788992223334445556667778889992222333344445557
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 start-time 2018-03-20.20:47
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 key-name 2345678922334455667788992223334445556667778889992222333344445558
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 start-time 2018-03-20.20:49
```

Use the `prompt` command to enter a secret key value. For example, the secret key value is 23456789223344556677889922233341234567892233445566778899222333 41.

```
[edit]
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 5 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 6 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 7 secret
```

```
New cak (secret):
Retype new cak (secret):
```

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

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 pre-shared-key-chain macsec-kc1
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 offset 50
crypto-officer@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]
crypto-officer@hostname:fips# set security macsec traceoptions file MACsec.log
crypto-officer@hostname:fips# set security macsec traceoptions file size 4000000000
crypto-officer@hostname:fips# set security macsec traceoptions flag all
```

5. Assign the trace to an interface.

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

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

To configure MACsec with keychain for ICMP traffic:

In R1:

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

```
[edit]
crypto-officer@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.

```
[edit]
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 key-name 2345678922334455667788992223334445556667778889992222333444455 1
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 start-time 2018-03-20.20:35
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 key-name 2345678922334455667788992223334445556667778889992222333444455 2
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 start-time 2018-03-20.20:37
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 key-name 2345678922334455667788992223334445556667778889992222333444455 3
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 start-time 2018-03-20.20:39
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 key-name 2345678922334455667788992223334445556667778889992222333444455 4
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 start-time 2018-03-20.20:41
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 key-name 2345678922334455667788992223334445556667778889992222333444455 5
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 start-time 2018-03-20.20:43
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 key-name 2345678922334455667788992223334445556667778889992222333444455 6
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 start-time 2018-03-20.20:45
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 key-name 2345678922334455667788992223334445556667778889992222333444455 7
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 start-time 2018-03-20.20:47
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 key-name 2345678922334455667788992223334445556667778889992222333444455 8
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 start-time 2018-03-20.20:49
```

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

```
[edit]
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 0 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 1 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 5 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 6 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 7 secret
New cak (secret):
Retype new cak (secret):
```

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

```
[edit]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 pre-shared-
```

```
key-chain macsec-kc1
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 offset 50
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 cipher-suite
gcm-aes-256
```

4. Set the trace option values.

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

5. Assign the trace to an interface.

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

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

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

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

4. Set the trace option values.

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

5. Assign the trace to an interface.

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

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

12. Configure bridge domain.

```
[edit]
crypto-officer@hostname:fips# set bridge-domains BD-110 domain-type bridge
crypto-officer@hostname:fips# set bridge-domains BD-110 vlan-id 100
```

```
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name1 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name2 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]
crypto-officer@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]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 pre-shared-
key-chain macsec-kc1
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 offset 50
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 cipher-suite
gcm-aes-256
```

3. Set the trace option values.

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

4. Assign the trace to an interface.

```
[edit]
crypto-officer@hostname:fips# set security macsec interfaces interface-name traceoptions
file mka_xe size 1g
```

```
crypto-officer@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]
crypto-officer@hostname:fips# set security macsec connectivity-association CA1 security-
mode static-cak
```

6. Set the MKA key server priority.

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

7. Set the MKA transmit interval.

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

8. Enable the MKA secure.

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

9. Assign the connectivity association to an interface.

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

10. Configure VLAN tagging.

```
[edit]
crypto-officer@hostname:fips# set interfaces interface-name1 flexible-vlan-tagging
crypto-officer@hostname:fips# set interfaces interface-name1 encapsulation flexible-
```

```
ethernet-services
crypto-officer@hostname:fips# set interfaces interface-name1 unit 100 encapsulation vlan-
bridge
crypto-officer@hostname:fips# set interfaces interface-name1 unit 100 vlan-id 100
crypto-officer@hostname:fips# set interfaces interface-name2 flexible-vlan-tagging
crypto-officer@hostname:fips# set interfaces interface-name2 encapsulation flexible-
ethernet-services
crypto-officer@hostname:fips# set interfaces interface-name2 unit 100 encapsulation vlan-
bridge
crypto-officer@hostname:fips# set interfaces interface-name2 unit 100 vlan-id 100
```

11. Configure bridge domain.

```
[edit]
crypto-officer@hostname:fips# set bridge-domains BD-110 domain-type bridge
crypto-officer@hostname:fips# set bridge-domains BD-110 vlan-id 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name1 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name2 100
```

Configuring MACsec with keychain for Layer 2 Traffic

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]
crypto-officer@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.

```
[edit]
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
```

```
key 0 start-time 2018-03-20.20:35
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 key-name 2345678922334455667788992223334445556667778889992222333444455552
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 start-time 2018-03-20.20:37
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 key-name 2345678922334455667788992223334445556667778889992222333444455553
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 start-time 2018-03-20.20:39
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 key-name 2345678922334455667788992223334445556667778889992222333444455554
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 start-time 2018-03-20.20:41
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 key-name 2345678922334455667788992223334445556667778889992222333444455555
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 start-time 2018-03-20.20:43
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 key-name 2345678922334455667788992223334445556667778889992222333444455556
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 start-time 2018-03-20.20:45
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 key-name 2345678922334455667788992223334445556667778889992222333444455557
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 start-time 2018-03-20.20:47
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 key-name 2345678922334455667788992223334445556667778889992222333444455558
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 start-time 2018-03-20.20:49
```

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

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

```
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 5 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 6 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 7 secret
New cak (secret):
Retype new cak (secret):
```

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

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

4. Set the trace option values.

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

5. Assign the trace to an interface.

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

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

12. Configure bridge domain.

```
[edit]
crypto-officer@hostname:fips# set bridge-domains BD-110 domain-type bridge
crypto-officer@hostname:fips# set bridge-domains BD-110 vlan-id 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name1 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name2 100
```

In R1:

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

```
[edit]
crypto-officer@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.

```
[edit]
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
```

```
key 0 start-time 2018-03-20.20:35
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 key-name 2345678922334455667788992223334445556667778889992222333444455 52
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 1 start-time 2018-03-20.20:37
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 key-name 2345678922334455667788992223334445556667778889992222333444455 53
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 2 start-time 2018-03-20.20:39
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 key-name 2345678922334455667788992223334445556667778889992222333444455 54
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 3 start-time 2018-03-20.20:41
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 key-name 2345678922334455667788992223334445556667778889992222333444455 55
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 4 start-time 2018-03-20.20:43
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 key-name 2345678922334455667788992223334445556667778889992222333444455 56
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 5 start-time 2018-03-20.20:45
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 key-name 2345678922334455667788992223334445556667778889992222333444455 57
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 6 start-time 2018-03-20.20:47
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 key-name 2345678922334455667788992223334445556667778889992222333444455 58
crypto-officer@hostname:fips#  set security authentication-key-chains key-chain macsec-kc1
key 7 start-time 2018-03-20.20:49
```

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

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

```
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 2 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 3 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 4 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 5 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 6 secret
New cak (secret):
Retype new cak (secret):
crypto-officer@hostname:fips#  prompt security authentication-key-chains key-chain macsec-
kc1 key 7 secret
New cak (secret):
Retype new cak (secret):
```

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

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

4. Set the trace option values.

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

5. Assign the trace to an interface.

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

7. Set the MKA key server priority.

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

8. Set the MKA transmit interval.

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

9. Enable the MKA secure.

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

10. Assign the connectivity association to an interface.

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

11. Configure VLAN tagging.

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

12. Configure bridge domain.

```
[edit]
crypto-officer@hostname:fips# set bridge-domains BD-110 domain-type bridge
crypto-officer@hostname:fips# set bridge-domains BD-110 vlan-id 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name1 100
crypto-officer@hostname:fips# set bridge-domains BD-110 interface interface-name2 100
```

CHAPTER 6

Configuring Event Logging

Event Logging Overview | 67

Configuring Event Logging to a Local File | 68

Interpreting Event Messages | 68

Logging Changes to Secret Data | 70

Login and Logout Events Using SSH | 70

Logging of Audit Startup | 71

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 following events:

- 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.

In addition, Juniper Networks recommends that logging also:

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

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 **Audit-File**:

```
[edit system]
syslog {
    file Audit-File;
}
```

Interpreting Event Messages

The following output shows a sample event message.

```
Feb 27 02:33:04 bm-a mgd[6520]: UI_LOGIN_EVENT: User 'security-officer' login, class 'j-super-
user' [6520],
ssh-connection '', client-mode 'cli'
Feb 27 02:33:49 bm-a mgd[6520]: UI_DBASE_LOGIN_EVENT: User 'security-officer' entering
configuration
mode
Feb 27 02:38:29 bm-a mgd[6520]: UI_CMDLINE_READ_LINE: User 'security-officer', command 'run show
log
Audit_log | grep LOGIN
```

Table 4 on page 69 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.	`Feb 27 02:33:04` is the timestamp expressed as local time in the United States. `2012-02-27T09:17:15.719Z` is `2:33 AM UTC on 27 Feb 2012.`
hostname	Name of the host that originally 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 | 67

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]
```

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

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 | 68

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
```

CHAPTER 7

Performing Self-Tests on a Device

Understanding FIPS Self-Tests | 73

Example: Configure FIPS Self-Tests | 73

Understanding FIPS Self-Tests

The cryptographic module enforces security rules to ensure that the Juniper Networks Junos operating system (Junos OS) in FIPS mode meets the security requirements of FIPS 140-2 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_kats`—KAT for kernel cryptographic routines
- `md_kats`—KAT for libmd and libc
- `openssl_kats`—KAT for OpenSSL cryptographic implementation
- `quicksec_kats`—KAT for QuickSec Toolkit cryptographic implementation
- `ssh_ipsec_kats`—KAT for SSH IPsec Toolkit cryptographic implementation
- `macsec_kats`—KAT for MACsec cryptographic implementation

The KAT self-tests are performed automatically at startup. Conditional self-tests are also performed automatically to verify digitally signed software packages, generated random numbers, RSA and ECDSA key pairs, and manually entered keys.

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

If there is KAT failure, 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.

You can also run FIPS self-test by issuing `request vmhost reboot` command. You can see the FIPS self-test logs on the console when the system is coming up.

Example: Configure FIPS Self-Tests

IN THIS SECTION

Hardware and Software Requirements | 74

Overview | 74
Configuration | 75
Verification | 76

This example shows how to configure FIPS self-tests to run periodically.

Hardware and Software Requirements

- You must have administrative privileges to configure FIPS self-tests.
- The device must be running the evaluated version of Junos OS in FIPS mode software.

Overview

The FIPS self-test consists of the following suites of known answer tests (KATs):

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

In this example, the FIPS self-test is executed at 9:00 AM in New York City, USA, every Wednesday.

NOTE: Instead of weekly tests, you can configure monthly tests by including the `month` and `day-of-month` statements.

When a KAT self-test fails, a log message is written to the system log messages file with details of the test failure. Then the system panics and reboots.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 75
- Step-by-Step Procedure | 75
- Results | 76

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

```
set system fips self-test periodic start-time 09:00
set system fips self-test periodic day-of-week 3
```

Step-by-Step Procedure

To configure the FIPS self-test, login to the device with crypto-officer credentials:

1. Configure the FIPS self-test to execute at 9:00 AM every Wednesday.

```
[edit system fips self-test]
crypto-officer@hostname:fips# set periodic start-time 09:00
crypto-officer@hostname:fips# set periodic day-of-week 3
```

2. If you are done configuring the device, commit the configuration.

```
[edit system fips self-test]
crypto-officer@hostname:fips# commit
```

Results

From configuration mode, confirm your configuration by issuing the `show system` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
crypto-officer@hostname:fips# show system
fips {
    self-test {
        periodic {
            start-time "09:00";
            day-of-week 3;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the FIPS Self-Test | 76

Confirm that the configuration is working properly.

Verifying the FIPS Self-Test

IN THIS SECTION

- Purpose | 77
- Action | 77
- Meaning | 80

Purpose

Verify that the FIPS self-test is enabled.

Action

Run the FIPS self-test manually by issuing the `request system fips self-test` command or reboot the device.

After issuing the request system fips self-test command or reboot the device, the system log file is updated to display the KATs that are executed. To view the system log file, issue the `file show /var/log/messages` command.

```
user@host# file show /var/log/messages
RE KATS:
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 KATS:
mgd:   NIST 800-90 HMAC DRBG Known Answer Test:         Passed
mgd:   FIPS ECDSA Known Answer Test:                    Passed
mgd:   FIPS ECDH Known Answer Test:                     Passed
mgd:   FIPS RSA Known Answer Test:                      Passed
mgd:   DES3-CBC Known Answer Test:                      Passed
mgd:   HMAC-SHA1 Known Answer Test:                     Passed
```

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

```
mgd:   HMAC-SHA1 Known Answer Test:                    Passed
mgd:   HMAC-SHA2-256 Known Answer Test:                Passed
mgd:   AES-CBC Known Answer Test:                      Passed
mgd:   SSH-RSA-ENC Known Answer Test:                  Passed
mgd:   SSH-RSA-SIGN Known Answer Test:                 Passed
mgd:   KDF-IKE-V1 Known Answer Test:                   Passed
mgd: Testing file integrity:
mgd:   File integrity Known Answer Test:               Passed
mgd: Testing crypto integrity:
mgd:   Crypto integrity Known Answer Test:             Passed
mgd: Expect an exec AuthenticatiMAC/veriexec: no fingerprint (file=/sbin/kats/cannot-exec
fsid=246 fileid=49356 gen=1 uid=0 pid=9384 ppid=9354 gppid=9352)on error...
mgd: /sbin/kats/run-tests: /sbin/kats/cannot-exec: Authentication error
mgd: FIPS Self-tests Passed

LC KATS:

Sep 12 10:50:44  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/0:0: <LC: Slot
no> pic:0 port:0 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:50:50  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/1:0: <LC: Slot
no> pic:0 port:1 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:50:55  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/0:0: <LC: Slot
no> pic:0 port:0 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:50:56  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/2:0: <LC: Slot
no> pic:0 port:2 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:01  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/1:0: <LC: Slot
no> pic:0 port:1 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:02  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/2:0: <LC: Slot
no> pic:0 port:2 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:06  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/3:0: <LC: Slot
no> pic:0 port:3 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:12  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/3:0: <LC: Slot
no> pic:0 port:3 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:17  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/4:0: <LC: Slot
no> pic:0 port:4 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:17  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/4:0: <LC: Slot
no> pic:0 port:4 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:26  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/5:0: <LC: Slot
no> pic:0 port:5 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:27  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/5:0: <LC: Slot
no> pic:0 port:5 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:36  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/6:0: <LC: Slot
no> pic:0 port:6 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
```

```
Sep 12 10:51:36  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/6:0: <LC: Slot
no> pic:0 port:6 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:44  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/7:0: <LC: Slot
no> pic:0 port:7 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:44  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/7:0: <LC: Slot
no> pic:0 port:7 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:51  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/8:0: <LC: Slot
no> pic:0 port:8 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:51  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/8:0: <LC: Slot
no> pic:0 port:8 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:51:58  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/9:0: <LC: Slot
no> pic:0 port:9 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:51:58  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/9:0: <LC: Slot
no> pic:0 port:9 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:52:05  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/10:0: <LC:
Slot no> pic:0 port:10 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:52:05  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/10:0: <LC:
Slot no> pic:0 port:10 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:52:12  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/11:0: <LC:
Slot no> pic:0 port:11 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:52:12  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/0/11:0: <LC:
Slot no> pic:0 port:11 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:52:20  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/1/0:0: <LC: Slot
no> pic:1 port:0 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:52:20  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/1/0:0: <LC: Slot
no> pic:1 port:0 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:52:27  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/1/1:0: <LC: Slot
no> pic:1 port:1 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
Sep 12 10:52:28  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/1/1:0: <LC: Slot
no> pic:1 port:1 chan:0 FIPS AES-256-GCM MACsec KATS decryption passed
Sep 12 10:52:34  <DUT-NAME> <LC slot> network_macsec_kats_input xe-<LC slot no>/1/2:0: <LC: Slot
no> pic:1 port:2 chan:0 FIPS AES-256-GCM MACsec KATS encryption passed
```

Meaning

The system log file displays the date and the time at which the KATs were executed and their status.

8
CHAPTER

Operational Commands

request system zeroize | 82

request vmhost zeroize no-forwarding | 83

request system zeroize

IN THIS SECTION

- Syntax | 82
- Description | 82
- Required Privilege Level | 82
- Release Information | 83

Syntax

```
request system zeroize
```

Description

For RE1800, remove all configuration information on the Routing Engines and reset all key values. If the device has dual Routing Engines, the command is broadcast to all Routing Engines on the device. The command removes all data files, including customized configuration and log files, by unlinking the files from their directories. The command removes all user-created files from the system including all plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, IPsec, RADIUS, TACACS+, and SNMP.

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

Required Privilege Level

maintenance

Release Information

Command introduced in Junos OS Release 12.2.

request vmhost zeroize no-forwarding

IN THIS SECTION

- Syntax | 83
- Description | 83
- Required Privilege Level | 84
- Sample Output | 84
- Release Information | 85

Syntax

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

Description

For REMX2K-X8, remove all configuration information on the Routing Engines and reset all key values. If the device has dual Routing Engines, the command is broadcast to both Routing Engines on the device. The command removes all data files, including customized configuration and log files, by unlinking the files from their directories. The command removes all user-created files from the system including all plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, IPsec, RADIUS, TACACS+, and SNMP.

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

Required Privilege Level

maintenance

Sample Output

request vmhost zeroize no-forwarding

```
user@host> request vmhost zeroize no-forwarding

VMHost Zeroization : Erase all data, including configuration and log files ?
[yes,no] (no) yes
re0:
--------------------------------------------------------------------------
warning: Vmhost will reboot and may not boot without configuration
warning: Proceeding with vmhost zeroize
Zeroise secondary internal disk ...
Proceeding with zeroize on secondary disk
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of secondary disk completed
Zeroize primary internal disk ...
Proceeding with zeroize on primary disk
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_rsa_key.pub
/etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_dsa_key
Mounting device in preparation for zeroize...
Cleaning up target disk for zeroize ...
Zeroize done on target disk.
Zeroize of primary disk completed
Zeroize done
---(more)--- Stopping cron.
Waiting for PIDS: 6135.
.
Feb 16 14:59:33 jlaunchd: periodic-packet-services (PID 6181) terminate signal 15
```

```
sent
Feb 16 14:59:33 jlaunchd: smg-service (PID 6234) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: application-identification (PID 6236) terminate signal
15 sent
Feb 16 14:59:33 jlaunchd: ifstate-tracing-process (PID 6241) terminate signal 15
sent
Feb 16 14:59:33 jlaunchd: resource-management (PID 6243) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: charged (PID 6246) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: license-service (PID 6255) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: ntp (PID 6620) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: gkd-chassis (PID 6621) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: gkd-lchassis (PID 6622) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: routing (PID 6625) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: sonet-aps (PID 6626) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: remote-operations (PID 6627) terminate signal 15 sent
Feb 16 14:59:33 jlaunchd: class-of-service
........
99
```

Release Information

Command introduced in Junos OS Release 15.1F3.