JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

FIPS Evaluated Configuration Guide for NFX150 Network Services Platform

Published
2020-10-01

Release
19.2R1

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 NFX150 Network Services Platform
19.2R1
Copyright © 2020 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About the Documentation | viii

Documentation and Release Notes | **viii**

Documentation Conventions | **viii**

Documentation Feedback | **xi**

Requesting Technical Support | **xi**

- Self-Help Online Tools and Resources | **xii**
- Creating a Service Request with JTAC | **xii**

1 **Overview**

Understanding Junos OS in FIPS Mode | 14

- About the Cryptographic Boundary on Your Device | **14**
- How FIPS Mode Differs from Non-FIPS Mode | **14**
- Validated Version of Junos OS in FIPS Mode | **15**

Understanding FIPS Terminology and Supported Cryptographic Algorithms | 15

- Terminology | **15**
- Supported Cryptographic Algorithms | **17**

Identifying Secure Product Delivery | 18

Understanding FIPS Error States and System Panic | 19

- FIPS System Panic | **19**
- Error Recovery from Alternate Boot Media | **20**

Understanding Management Interfaces | 21

2 **Configuring Roles and Authentication Methods**

Understanding Roles and Services for Junos OS in FIPS Mode | 23

- Crypto Officer Role and Responsibilities | **23**
- FIPS User Role and Responsibilities | **24**

What Is Expected of All FIPS Users | 24

Understanding the Operational Environment for Junos OS in FIPS Mode | 25

Hardware Environment for Junos OS in FIPS Mode | 26

Software Environment for Junos OS in FIPS Mode | 26

Critical Security Parameters | 27

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 30

Downloading Software Packages from Juniper Networks | 31

Installing Software on Network Services Platform with a Single Routing Engine | 32

Understanding Zeroization to Clear System Data for FIPS Mode | 33

Why Zeroize? | 34

When to Zeroize? | 34

Zeroizing the System | 35

Establishing Root Password Access | 36

Enabling FIPS Mode | 37

Configuring Crypto Officer and FIPS User Identification and Access | 40

Configuring Crypto Officer Login Access | 41

Configuring FIPS User Login Access | 42

3 **Configuring Administrative Credentials and Privileges**

Understanding the Associated Password Rules for an Authorized Administrator | 46

Authentication methods in FIPS mode of Operation | 48

Username and Password Authentication over the Console and SSH | 48

Username and Public Key Authentication over SSH | 48

4 **Configuring SSH and Console Connection**

Configuring SSH on the Evaluated Configuration for FIPS | 51

Configuring the time and date | 52

Configuring the User Session Idle Timeout | 53

5 Configuring the Remote Syslog Server

Syslog Server Configuration on a Linux System | 55

Configuring Event Logging to a Local File | 55

Configuring Event Logging to a Remote Server | 55

Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server | 56

6 Configuring Event Logging

Event Logging Overview | 63

Configuring Event Logging to a Local File | 64

Interpreting Event Messages | 64

Logging Changes to Secret Data | 65

Login and Logout Events Using SSH | 67

Logging of Audit Startup | 68

7 Configuring VPNs

Configuring VPN on a Device Running Junos OS | 70

Configuring an IPsec VPN with a Preshared Key for IKE Authentication | 72

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Initiator | 73

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Responder | 76

Configuring an IPsec VPN with an RSA Signature for IKE Authentication | 79

Configuring IPsec VPN with RSA Signature as IKE Authentication on the Initiator | 80

Configuring IPsec VPN with RSA Signature as IKE Authentication on the Responder | 83

Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication | 86

Configuring IPsec VPN with ECDSA signature IKE authentication on the Initiator | 86

Configuring IPsec VPN with ECDSA signature IKE authentication on the Responder | 89

8 Configuring Security Flow Policies

Understanding a Security Flow Policy on a Device Running Junos OS | 94

Configuring a Security Flow Policy in Firewall Bypass Mode | 94

Configuring a Security Policy in Firewall Discard Mode | 95

Configuring a Security Flow Policy in IPsec Protect Mode | 95

9 Configuring Traffic Filtering Rules

Understanding Protocol Support | 98

Configuring Traffic Filter Rules | 99

Configuring Default Deny-All and Reject Rules | 100

Logging the Dropped Packets Using Default Deny-all Option | 101

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets | 102

Configuring Default Reject Rules for Source Address Spoofing | 103

Configuring Default Reject Rules with IP Options | 103

Configuring Default Reject Rules | 105

Configuring the Device to Drop Unassigned IPv6 Packets | 106

10 Configuring Network Attacks

Configuring IP Teardrop Attack Screen | 109

Configuring TCP Land Attack Screen | 110

Configuring ICMP Fragment Screen | 112

Configuring Ping-Of-Death Attack Screen | 113

Configuring tcp-no-flag Attack Screen | 115

Configuring TCP SYN-FIN Attack Screen | 116

Configuring TCP fin-no-ack Attack Screen | 118

Configuring UDP Bomb Attack Screen | 119

Configuring UDP CHARGEN DoS Attack Screen | 119

Configuring TCP SYN and RST Attack Screen | 121

Configuring ICMP Flood Attack Screen | 123

Configuring TCP SYN Flood Attack Screen | 124

Configuring TCP Port Scan Attack Screen | 126

Configuring UDP Port Scan Attack Screen | 127

Configuring IP Sweep Attack Screen | 129

11 Configuring the IDP Extended Package

IDP Extended Package Configuration Overview | 132

12 Performing Self-Tests on a Device

Understanding FIPS Self-Tests | 134

Example: Configuring FIPS Self-Tests | 134

Verifying That FIPS Self-Tests Are Taking Place | 136

13 Junos-FIPS Configuration Restrictions

Understanding Remote Access for Junos OS in FIPS Mode | 141

Understanding Configuration Limitations and Restrictions on Junos OS in FIPS Mode | 141

14 Configuration Statements

fips (FIPS) | 145

level (FIPS) | 146

chassis level (FIPS) | 147

15 Operational Commands

request system zeroize (FIPS) | 149

About the Documentation

IN THIS SECTION

- Documentation and Release Notes | viii
- Documentation Conventions | viii
- Documentation Feedback | xi
- Requesting Technical Support | xi

Use this guide to operate the NFX device 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.

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at https://www.juniper.net/documentation/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Documentation Conventions

Table 1 on page ix defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page ix defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

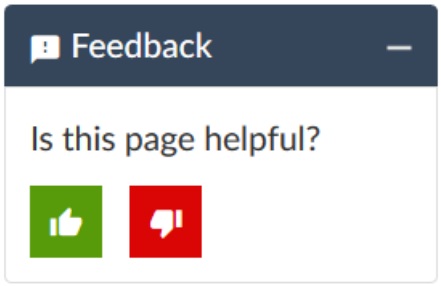
Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services support contract, or are

covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://entitlementsearch.juniper.net/entitlementsearch/

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

CHAPTER 1

Overview

Understanding Junos OS in FIPS Mode | **14**

Understanding FIPS Terminology and Supported Cryptographic Algorithms | **15**

Identifying Secure Product Delivery | **18**

Understanding FIPS Error States and System Panic | **19**

Understanding Management Interfaces | **21**

Understanding Junos OS in FIPS Mode

IN THIS SECTION

- About the Cryptographic Boundary on Your Device | 14
- How FIPS Mode Differs from Non-FIPS Mode | 14
- Validated Version of Junos OS in FIPS Mode | 15

Federal Information Processing Standards (FIPS) 140-2 defines security levels for hardware and software that perform cryptographic functions. By meeting the applicable overall requirements within the FIPS standard, the Juniper Networks NFX150 device running the Juniper Networks Junos operating system (Junos OS) in *FIPS mode* comply with the FIPS 140-2 Level 1 standard.

Operating NFX Series in a FIPS 140-2 Level 1 environment requires enabling and configuring FIPS mode on the NFX150 device from the Junos OS command-line interface (CLI).

The *Crypto Officer* enables FIPS mode and sets up keys and passwords for the system and other *FIPS users* who can view the configuration.

About the Cryptographic Boundary on Your Device

FIPS 140-2 compliance requires a defined *cryptographic boundary* around each *cryptographic module* on NFX150 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 by, for example, being displayed on a console or written to an external log file.

How FIPS Mode Differs from Non-FIPS Mode

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

- Self-tests of all cryptographic algorithms are performed at startup.
- Self-tests of random number and key generation are performed continuously.

- Weak cryptographic algorithms such as Data Encryption Standard (DES) and 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 advisor page on the Juniper Networks Web site (https://apps.juniper.net/compliance/).

RELATED DOCUMENTATION

Identifying Secure Product Delivery | 18

Understanding FIPS Terminology and Supported Cryptographic Algorithms

IN THIS SECTION

Terminology | 15
Supported Cryptographic Algorithms | 17

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 25.

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. NFX150 Network Services Platform is certified at FIPS 140-2 Level 1. For fixed-configuration NFX150 device, the cryptographic module is the NFX150 device case. For modular NFX150 device, the cryptographic module is the Routing Engine.

Crypto Officer—Person with appropriate permissions who is responsible for securely enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode on an NFX150 device. For details, see "Understanding Roles and Services for Junos OS in FIPS Mode" on page 23.

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 134.

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 an NFX150 device before its operation as a FIPS cryptographic module—or in preparation for repurposing the NFX150 device for non-FIPS operation. The Crypto Officer can zeroize the system with a CLI operational command.

Supported Cryptographic Algorithms

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.

3DES (3des-cbc)—Encryption standard based on the original Data Encryption Standard (DES) from the 1970s that used a 56-bit key and was cracked in 1997. The more secure 3DES is DES enhanced with three multiple stages and effective key lengths of about 112 bits. For Junos OS in FIPS mode, 3DES is implemented with cipher block chaining (CBC).

NOTE: 3DES is supported only in FIPS.

RELATED DOCUMENTATION

Understanding FIPS Self-Tests | 134

Understanding Zeroization to Clear System Data for FIPS Mode | 33

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 FIPS Error States and System Panic

IN THIS SECTION

- FIPS System Panic | 19
- Error Recovery from Alternate Boot Media | 20

An NFX150 device operating Junos OS in FIPS mode has certain operational restrictions such as the ability to load only integrity-checked software files and use only FIPS-approved cryptographic algorithms. To ensure correct operation, the NFX150 device performs a series of FIPS self-tests.

The NFX150 device performs additional tests as needed—for example, to ensure that randomly generated numbers are truly random and to verify manually entered keys (passwords).

If it fails a test, the NFX150 device enters a FIPS error state known as *system panic*.

FIPS errors that occur early in the boot cycle can prevent the system from successfully starting up. For this reason, keep alternate boot media up to date.

For details, see:

FIPS System Panic

If an NFX150 device fails a FIPS self-test, the NFX150 device enters a FIPS error state known as *system panic*. The panic condition halts all cryptographic processing and stops all data output from the NFX150 device. To clear the FIPS error, NFX150 device reboots, runs the FIPS self-tests, and if it passes all the tests, returns to normal operation.

If NFX150 device fails a self-test during a reboot from panic mode, the system stops booting and attempts to reboot. If the reboot is unsuccessful, the NFX150 device attempts again to reboot, this time from available boot media.

During a system panic, only status messages are displayed on the console. For example, a FIPS error is logged as follows:

```
panic: pid 5090 (fips-error), uid 0, FIPS error 5: cannot verify certificate
PackageCA
```

The reboot after panic displays the following error message on the console:

```
savecore: reboot after panic: pid 5090 (fips-error), uid 0, FIPS error 5: cannot
verify certificate PackageCA
```

The following error states create a system panic:

> **NOTE:** These errors have only an extremely small chance of occurring.

- The NFX150 device failed a known answer test (KAT).
- The random number is not random.
- Signature generation failed.
- Signature verification failed.
- Certificate verification failed.
- Encryption or decryption failed.
- An environment error occurred.
- An error occurred in a pair-wise conditional test.

Error Recovery from Alternate Boot Media

An NFX150 Network Services Platform running Junos OS in FIPS mode performs KATs self-tests at startup. If the NFX150 device fails a KAT, the boot process stops and the NFX150 device attempts to reboot. If the reboot is unsuccessful, the NFX150 device attempts again to reboot, this time from available boot media.

However, if the seal is broken, the NFX150 device is no longer a FIPS cryptographic module. You as Crypto Officer must zeroize, reinstall and reconfigure Junos OS and enable FIPS mode.

Understanding Management Interfaces

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

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

2

CHAPTER

Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS in FIPS Mode | **23**

Understanding the Operational Environment for Junos OS in FIPS Mode | **25**

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | **30**

Downloading Software Packages from Juniper Networks | **31**

Installing Software on Network Services Platform with a Single Routing Engine | **32**

Understanding Zeroization to Clear System Data for FIPS Mode | **33**

Zeroizing the System | **35**

Establishing Root Password Access | **36**

Enabling FIPS Mode | **37**

Configuring Crypto Officer and FIPS User Identification and Access | **40**

Understanding Roles and Services for Junos OS in FIPS Mode

IN THIS SECTION

- Crypto Officer Role and Responsibilities | 23
- FIPS User Role and Responsibilities | 24
- What Is Expected of All FIPS Users | 24

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.

In addition to their FIPS roles, both Crypto Officer and user can perform normal configuration tasks on the NFX150 device as individual user configuration allows.

Crypto Officers and FIPS users perform 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.

For details, see:

Crypto Officer Role and Responsibilities

The Crypto Officer is the person responsible for enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode on NFX Series. The Crypto Officer securely installs Junos OS on the NFX150 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 atleast 10 characters.
- Reset user passwords for FIPS-approved algorithms during upgrades from Junos OS.
- Set up manual IPsec SAs for configuration with dual Routing Engines.
- Examine log and audit files for events of interest.
- Erase user-generated files, 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 routers or switches and documentation in a secure area.
- Deploy routersor switches 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 | 35

Configuring Crypto Officer and FIPS User Identification and Access | 40

Understanding the Operational Environment for Junos OS in FIPS Mode

IN THIS SECTION

Hardware Environment for Junos OS in FIPS Mode | 26

Software Environment for Junos OS in FIPS Mode | 26

Critical Security Parameters | 27

An NFX150 Network Services Platform 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 NFX150 device in non-FIPS mode:

Hardware Environment for Junos OS in FIPS Mode

Junos OS in FIPS mode establishes a cryptographic boundary in the NFX150 device that no critical security parameters (CSPs) can cross using plain text. Each hardware component of the NFX150 device that requires a cryptographic boundary for FIPS 140-2 compliance is a separate cryptographic module.

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 NFX150 device running Junos OS in FIPS mode forms a special type of nonmodifiable operational environment. To achieve this environment on the NFX150 device , the system prevents the execution of any binary file that was not part of the certified Junos OS in FIPS mode distribution. When an NFX150 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 Release image that includes FIPS package 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.

Installing the Junos-FIPS image 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.

NOTE: In Junos OS in FIPS mode, user authentication data can be entered in plain text. Otherwise, all CSPs must enter and leave the cryptographic module in encrypted form. In general, configuration should be done with SSH.

NOTE: RADIUS and TACACS+ authentication functionalities are disabled for Junos OS in FIPS mode.

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). The default password format must be changed to SHA256, or SHA512. 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 device or Routing Engine as a cryptographic module.

Table 3 on page 28 lists CSPs on routers running Junos OS.

Table 3: Critical Security Parameters

CSP	Description	Zeroization method	Use
SSH-2 private host key	ECDSA key used to identify the host, generated the first time SSH is configured. RSA key used to identify the host, generated the first time SSH is configured.	Zeroize command.	Used to identify the host.
SSH-2 session key	Session key used with SSH-2. and as a Diffie-Hellman private key. Encryption: 3DES (FIPS only), AES-128, AES-256. MACs: HMAC-SHA-1, HMAC SHA-256, HMAC SHA-512. Key exchange: DH Group exchange (2048 ≤ key ≤ 8192), ECDH: ECDH-sha2-nistp256, ECDH-sha2-nistp384, and ECDH-sha2-nistp521.	Power cycle and terminate session.	Symmetric key used to encrypt data between host and client.
User authentication key	Hash of the user's password: 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.
HMAC DRBG key value	The current value of the outlen-bit key, which is updated at least once each time that the DRBG mechanism generates pseudorandom bits.	Power cycle.	A critical value of the internal state of DRBG.

Table 3: Critical Security Parameters *(continued)*

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

NOTE: 3DES is not claimed for NDcPPv2.

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. However, as the Crypto Officer, you can enter user authentication data in plain text.

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.

RELATED DOCUMENTATION

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 30

Understanding Zeroization to Clear System Data for FIPS Mode | 33

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode

Ensure that the NFX150 device is in FIPS mode before you configure the Crypto Officer or any users. 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—for example, 20 characters for SHA-1 authentication. .

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

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

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

- Words that might be found in or exist as a permuted form in a system files such as **/etc/passwd**.
- The hostname of the system (always a first guess).
- Any word or phrase that appears in a dictionary or other well-known source, including dictionaries and thesauruses in languages other than English; works by classical or popular writers; or common words and phrases from sports, sayings, movies or television shows.

- Permutations on any of the above—for example, a dictionary word with letters replaced with digits (**root**) or with digits added to the end.
- Any machine-generated password. Algorithms reduce the search space of password-guessing programs and so must not be used.

RELATED DOCUMENTATION

Understanding the Operational Environment for Junos OS in FIPS Mode | **25**

Downloading Software Packages from Juniper Networks

You can download the Junos OS software packages 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/entitlement/setupAccountInfo.do.

To download software packages from Juniper Networks:

1. Using a Web browser, follow the links to the download URL on the Juniper Networks webpage.

   https://www.juniper.net/support/downloads/junos.html

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. Select the appropriate software package:

   - For Junos OS package, ensure that the name contains the correct device name and number of the Junos OS release that is FIPS-certified on NFX150 Network Services Platform.

     For example, **jinstall-host-nfx-3-x86-64-19.2R1.4-secure-signed.tgz**.

4. Download the software to a local host or to an internal software distribution site.

5. Install the Junos OS. See “Installing Software on Network Services Platform with a Single Routing Engine” on page 32.

Installing Software on Network Services Platform with a Single Routing Engine

You can use this procedure to upgrade Junos OS on an NFX150 Network Services Platform with a single Routing Engine.

> **NOTE:** Junos OS is delivered in signed packages that contain digital signatures to ensure the Juniper Networks software is running. When installing the software packages, Junos OS validates the signatures and the public key certificates used to digitally sign the software packages. If the signature or certificate is found to be invalid (for example, when the certificate validity period has expired or cannot be verified against the root CA stored in the Junos OS internal store), the installation process fails.

To install software upgrades on an NFX150 device with a single Routing Engine:

1. Download the software package as described in "Downloading Software Packages from Juniper Networks" on page 31.

2. If you have not already done so, connect to the console port on the NFX150 device from your management device, and log in to the CLI. (For instructions, see Initial Configuration on NFX150 Devices.)

3. (Optional) Back up the current software configuration to a second storage option. See the *Junos OS Installation and Upgrade Guide* for instructions on performing this task.

4. (Optional) Copy the software package to the NFX150 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 NFX150 device:

   ```
   user@host-name> request vmhost software add
   ```

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

   - For a software package in a local directory on the NFX150 device, use **/var/tmp/*package*.tgz**.
   - **ftp://*hostname*/*pathname*/*package*.tgz**

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

6. Reboot the device to load the installation and start the new software:

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

Understanding Zeroization to Clear System Data for FIPS Mode

IN THIS SECTION

- Why Zeroize? | 34
- When to Zeroize? | 34

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

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

Use **delete system-phone home** command to delete all phone-home related configurations.

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 switch for operation as a FIPS cryptographic module, perform zeroization before enabling FIPS mode.
- **Before disabling FIPS mode of operation:** To begin repurposing your switch for non-FIPS mode of operation, perform zeroization on the switch.

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

RELATED DOCUMENTATION

Zeroizing the System | 35

Enabling FIPS Mode | 37

Zeroizing the System

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

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

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

To zeroize your device:

CAUTION: Perform system zeroization with care. After the zeroization process is complete, no data is left on the Routing Engine. The device is returned to the factory default state, without any configured users or configuration files. If internal IPSec tunnel is configured, both REs will be zeroized.

1. From the CLI, enter

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

RELATED DOCUMENTATION

Enabling FIPS Mode | 37

Understanding Zeroization to Clear System Data for FIPS Mode | 33

Establishing Root Password Access

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 30. 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 SHA-1, SHA-256 or SHA-512. Password recovery is not possible in Junos OS in FIPS mode. Junos OS in FIPS mode cannot boot into single-user mode without the correct root password.

After you log in, configure the root (superuser) password to be used to access the NFX150 device as follows:

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

   ```
   {master:0}
   host-name> configure
   Entering configuration mode
   ```

2. Configure the root password by including the **root-authentication** statement at the **[edit system]** hierarchy level and selecting one of the password options.

   - To configure a plain-text password, select the **plain-text-password** option. Enter and confirm the password at the prompts.

     ```
     {master:0}
     [edit system ]
     host-name#set root-authentication plain-text-password
     New password: type password here
     Retype new password: retype password here
     ```

     Ensure that you follow the password guidelines in "Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 30.

   - To configure public keys for SSH authentication of root logins, use the **ssh-ecdsa** option. You can configure more than one public key for SSH authentication of root logins and for user accounts.

When a user logs in as **root**, the public keys are referenced to determine whether the private key matches any of them.

3. If you are finished configuring the NFX150 device, commit the configuration and quit:

```
{master:0}
[edit]
host-name# commit
commit complete
host-name# quit
```

RELATED DOCUMENTATION

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 30

Enabling FIPS Mode

When Junos OS is installed on NFX150 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 30. 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 33 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 19.2-20180131.0 Kernel 64-bit  JNPR-11.0-20180123.155949_fbsd-
 root@:~ # cli
root>
```

3. Configure root authentication.

```
 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.

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

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

6. Install **jpfe-fips** package.

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

7. • Configure chassis boundary fips by setting **set system fips chassis level 1** and **commit**.
   • Configure 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 older CSP in loaded configuration.

8. After deleting and reconfiguring CSPs, commit will go through and device needs reboot to enter FIPS mode.

```
 [edit]
 root@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
```

request vmhost reboot

```
root@hostname# request vmhost reboot
```

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

```
root@hostname:fips>
```

10. After the reboot has completed, log in and use the **show version** command to verify.

```
user@device>  show version
Hostname: nfx
Model: nfx150_s1e
Junos: 19.2-20180423_19.2T_x_tvp.0
JUNOS OS Kernel 64-bit  [20180413.173511_fbsd-builder_stable_11]
JUNOS OS libs [20180413.173511_fbsd-builder_stable_11]
JUNOS OS runtime [20180413.173511_fbsd-builder_stable_11]
JUNOS OS time zone information [20180413.173511_fbsd-builder_stable_11]
JUNOS network stack and utilities [20180423.075043_dc-builder_release_192_throttle]
JUNOS libs [20180423.075043_dc-builder_release_192_throttle]
JUNOS OS libs compat32 [20180413.173511_fbsd-builder_stable_11]
JUNOS OS 32-bit compatibility [20180413.173511_fbsd-builder_stable_11]
JUNOS libs compat32 [20180423.075043_dc-builder_release_192_throttle]
JUNOS runtime [20180423.075043_dc-builder_release_192_throttle]
JUNOS Packet Forwarding Engine Simulation Package
[20180423.075043_dc-builder_release_192_throttle]
JUNOS py extensions [20180423.075043_dc-builder_release_192_throttle]
JUNOS py base [20180423.075043_dc-builder_release_192_throttle]
JUNOS OS vmguest [20180413.173511_fbsd-builder_stable_11]
JUNOS OS crypto [20180413.173511_fbsd-builder_stable_11]
JUNOS Wireless WAN Module [20180423.075043_dc-builder_release_192_throttle]
Junos vmguest package [20180423.075043_dc-builder_release_192_throttle]
JUNOS Unified Threat Management Module
[20180423.075043_dc-builder_release_192_throttle]
JUNOS userfw [20180423.075043_dc-builder_release_192_throttle]
JUNOS syshmd [20180423.075043_dc-builder_release_192_throttle]
```

```
JUNOS switch CLI for NFX-3 [20180423.075043_dc-builder_release_192_throttle]
JUNOS security base [20180423.075043_dc-builder_release_192_throttle]
JUNOS mx libs compat32 [20180423.075043_dc-builder_release_192_throttle]
JUNOS mx runtime [20180423.075043_dc-builder_release_192_throttle]
JUNOS pppoe [20180423.075043_dc-builder_release_192_throttle]
JUNOS common platform support [20180423.075043_dc-builder_release_192_throttle]
JUNOS nfx platform support [20180423.075043_dc-builder_release_192_throttle]
JUNOS mtx network modules [20180423.075043_dc-builder_release_192_throttle]
JUNOS named module [20180423.075043_dc-builder_release_192_throttle]
JUNOS modules [20180423.075043_dc-builder_release_192_throttle]
JUNOS mx modules [20180423.075043_dc-builder_release_192_throttle]
JUNOS mx libs [20180423.075043_dc-builder_release_192_throttle]
JUNOS jdm ure cmd [20180423.075043_dc-builder_release_192_throttle]
JUNOS jdm cmd [20180423.075043_dc-builder_release_192_throttle]
JUNOS idpd [20180423.075043_dc-builder_release_192_throttle]
JUNOS nfx Data Plane Crypto Support
[20180423.075043_dc-builder_release_192_throttle]
JUNOS daemons [20180423.075043_dc-builder_release_192_throttle]
JUNOS mx daemons [20180423.075043_dc-builder_release_192_throttle]
JUNOS appidd application-identification
daemon[20180423.075043_dc-builder_release_192_throttle]
JUNOS Extension Toolkit [20180423.075043_dc-builder_release_192_throttle]
JUNOS Phone-home [20180423.075043_dc-builder_release_192_throttle]
JUNOS Packet Forwarding Engine FIPS Support [19.2-20180423_19.2T_x_tvp.0]
JUNOS J-Insight [20180423.075043_dc-builder_release_192_throttle]
JUNOS Online Documentation [20180423.075043_dc-builder_release_192_throttle]
JUNOS jail runtime [20180413.173511_fbsd-builder_stable_11]
JUNOS FIPS mode utilities [20180423.075043_dc-builder_release_192_throttle]
```

Configuring Crypto Officer and FIPS User Identification and Access

IN THIS SECTION

- Configuring Crypto Officer Login Access | **41**
- Configuring FIPS User Login Access | **42**

Crypto Officers and FIPS users perform 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 Login 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:

```
{master:0}
  root:fips> configure
     Entering configuration mode
{master:0} [edit]
root:fips#
```

2. Name the user "crypto-officer" and assign the Crypto Officer a user ID (for example, **6400**) and a class (for example, **super-user**). When you assign the class, you assign the permissions—for example, **secret**, **security**, **maintenance**, and **control**.

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

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

3. Following the guidelines in "Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 30, 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:fips# set system login user crypto-officer class super-user authentication plain-text-password
```

4. Optionally, display the configuration:

```
[edit]
root:fips# edit system
[edit system]
root:fips# show
login {
    user crypto-officer {
        uid 6400;
        authentication {
            encrypted-password "<cipher-text>"; ## SECRET-DATA
        }
        class super-user;
    }
}
```

5. If you are finished configuring the NFX150 device, commit the configuration and exit:

```
[edit]
root:fips# commit
commit complete
root:fips# exit
root:fips> exit
```

Otherwise, go on to "Configuring FIPS User Login Access" on page 42.

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.

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

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

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

   ```
   [edit]
   crypto-officer:fips# set system login user fips-user1 uid 6401 class operator
   ```

3. Following the guidelines in "Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode" on page 30, assign the FIPS a plain-text password for login authentication. Set the password by typing a password after the prompts **New password** and **Retype new password**.

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

4. Optionally, display the configuration:

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

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

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

   Otherwise, go on to *Configuring the Console Port for FIPS Mode*.

RELATED DOCUMENTATION

- Understanding Roles and Services for Junos OS in FIPS Mode | 23

3

CHAPTER

Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator | **46**

Authentication methods in FIPS mode of Operation | **48**

Understanding the Associated Password Rules for an Authorized Administrator

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

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

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

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

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

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

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

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

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

NOTE: Configure the hashing algorithm used for password storage as sha512.

```
[ 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.

NOTE: Passwords should be changed periodically.

RELATED DOCUMENTATION

Identifying Secure Product Delivery | **18**

Authentication methods in FIPS mode of Operation

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

- Username and Password Authentication over the Console and SSH on page 48
- Username and Public Key Authentication over SSH on page 48

Username and Password Authentication over the Console and SSH

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

> **NOTE:** The maximum password length is 20 characters.

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

Therefore, this leads to a maximum of seven possible attempts in a 1-minute period for each getty active terminal. So, the best approach for the attacker would be to disconnect after 4 failed attempts, and wait for a new getty to be spawned. This would allow the attacker to perform roughly 9.6 attempts per minute (576 attempts per hour or 60 minutes). This would be rounded off to 9 attempts per minute, because there is no such thing as 0.6 attempts. Thus the probability of a successful random attempt is 1/96^10, which is less than 1/1 million. The probability of a success with multiple consecutive attempts in a 1-minute period is 9/(96^10), which is less than 1/100,000.

Username and Public Key Authentication over SSH

With SSH public-key authentication, the user provides the username and proves ownership of the private key corresponding to the public key stored on the server. The device supports ECDSA (P-256, P-384, and

P-521) and RSA (2048-bit or higher since our RSA implementation is FIPS 186-4 compliant). The probability of a success with multiple consecutive attempts in a 1-minute period is 5.6e7/(2^112).

CHAPTER 4

Configuring SSH and Console Connection

Configuring SSH on the Evaluated Configuration for FIPS | 51

Configuring the time and date | 52

Configuring the User Session Idle Timeout | 53

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 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
```

Configuring the time and date

To configure a system date and time, use the following command:

```
[edit]
user@host# set date YYYYMMDDHHMM.ss
```

Configuring the User Session Idle Timeout

To configure the idle timeout for a user session, use the following command:

```
[edit]
user@host# set system login idle-timeout minutes
```

5
CHAPTER

Configuring the Remote Syslog Server

Syslog Server Configuration on a Linux System | 55

Syslog Server Configuration on a Linux System

A secure Junos OS environment requires auditing of events and storing them in a local audit file. The recorded events are simultaneously sent to an external syslog server. A syslog server receives the syslog messages streamed from the NFX150 device. The syslog server must have an SSH client with NETCONF support configured to receive the streamed syslog messages.

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

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

Configuring Event Logging to a Local File

Configure audit information to be stored in a local file on the device along with the level of detail using the "syslog" statement. The following must be used to ensure all events detailed in the NDcPP are logged and are stored in a local file named Audit_file in the following example:

```
[edit system]
syslog {
file Audit_file {
any any;
}
}
```

Configuring Event Logging to a Remote Server

Configure the export of audit information to a secure, remote server by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server.

The following procedures show the configuration needed to send system log messages from TOE to a secure external server by using NETCONF over SSH.

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

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

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

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

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

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

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

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

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

4. Set up NETCONF with SSH.

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

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

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

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

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

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

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

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

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

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

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

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

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

The following output shows test log results for syslog server.

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

Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/host/.ssh/syslog-monitor.
Your public key has been saved in /home/host/.ssh/syslog-monitor.pub.
The key fingerprint is:
ef:75:d7:68:c5:ad:8d:6f:5e:7a:7e:9b:3d:f1:4d:3f syslog-monitor key pair
The key's randomart image is:
```

```
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|               ..|
|        S       +|
|         .     Bo|
|          . . *.X|
|         . . o E@|
|          .   .BX|
+-----------------+
[host@nms5-vm-linux2 ~]$ cat /home/host/.ssh/syslog-monitor.pub
ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCrUREJUBpjwAoIgRrGy9zgt+
D2pikk3Q/Wdf8I5vr+njeqJhCx2bUAkrRbYXNILQQAZbg7kLfi/8TqqL
eon4HOP2e6oCSorKdx/GrOTzLONL4fh0EyuSAk8bs5JuwWNBUokV025
gzpGFsBusGnlj6wqqJ/sjFsMmfxyCkbY+pUWb8m1/A9YjOFT+6esw+9S
tF6Gbg+VpbYYk/Oday4z+z7tQHRFNfxj2G92aoliVDBLJparEMBc8w
LdSUDxmgBTM2oadOmm+kreBUQjrmr6775RJn9H9YwIxKOxGm4SFnX/Vl4
R+lZ9RqmKH2wodIEM34K0wXEHzAzNZ01oLmaAVqT
syslog-monitor key pair
[host@nms5-vm-linux2 ~]$ eval `ssh-agent`
Agent pid 1453
[host@nms5-vm-linux2 ~]$ ssh-add ~/.ssh/syslog-monitor
Enter passphrase for /home/host/.ssh/syslog-monitor:
Identity added: /home/host/.ssh/syslog-monitor (/home/host/.ssh/syslog-monitor)
```

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

<!-- No zombies were killed during the creation of this user interface --
<!-- user syslog-mon, class j-monitor -><hello>
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>
   <capability>urn:ietf:params:xml:ns:netconf:capability:candidate:1.0</capability>


<capability>urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0</capability>

   <capability>urn:ietf:params:xml:ns:netconf:capability:validate:1.0</capability>
```

```
<capability>urn:ietf:params:xml:ns:netconf:capability:url:1.0?protocol=http,ftp,file</capability>

    <capability>http://xml.juniper.net/netconf/junos/1.0</capability>
    <capability>http://xml.juniper.net/dmi/system/1.0</capability>
  </capabilities>
  <session-id4129/session-id>
</hello>
]]>]]>
```

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

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

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

```
Local : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation
in progress: Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd',
 PID 4317, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Dynamic flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/dfcd', PID 4318, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/cfmd', PID 4319, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
```

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

```
Remote : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation
 in progress: Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rdd',
 PID 4317, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Dynamic flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/dfcd', PID 4318, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Connectivity fault management process checking new configuration
```

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

Configuring Event Logging

Event Logging Overview | **63**

Configuring Event Logging to a Local File | **64**

Interpreting Event Messages | **64**

Logging Changes to Secret Data | **65**

Login and Logout Events Using SSH | **67**

Logging of Audit Startup | **68**

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.
- Changes to modification or deletion of cryptographic keys.
- Password resets.

In addition, Juniper Networks recommends that logging also:

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

RELATED DOCUMENTATION

Interpreting Event Messages | 64

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;
}
```

RELATED DOCUMENTATION

Event Logging Overview | 63

Interpreting Event Messages

The following output shows a sample event message.

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

Table 4 on page 64 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.	Apr 24 17:43:28 is the timestamp expressed as local time in the United States. 2018-04-24T09:17:15.719Z is 9:17 AM UTC on 24 April 2018.

Table 4: Fields in Event Messages *(continued)*

Field	Description	Examples
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 | 63

Logging Changes to Secret Data

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

Load Merge

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

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

Load Replace

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

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

Load Override

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

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

Load Update

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

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

```
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate:
[system radius-server 1.2.3.4 secret] ""
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set:
[system login user admin authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate:
[system login user admin authentication encrypted-password] ""
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set:
[system login user test authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate:
[system login user test authentication encrypted-password] ""
```

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

RELATED DOCUMENTATION

Interpreting Event Messages | 64

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 | 64

Logging of Audit Startup

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

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

RELATED DOCUMENTATION

Login and Logout Events Using SSH | 67

CHAPTER 7

Configuring VPNs

Configuring VPN on a Device Running Junos OS | 70

Configuring VPN on a Device Running Junos OS

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

- Configuring an IPsec VPN with a Preshared Key for IKE Authentication on page 72
- Configuring an IPsec VPN with an RSA Signature for IKE Authentication on page 79
- Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication on page 86

Figure 1 on page 70 illustrates the VPN topology used in all the examples described in this section. Here, H0 and H1 are the host PCs, R0 and R2 are the two endpoints of the IPsec VPN tunnel, and R1 is a router to route traffic between the two different networks.

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

Figure 1: VPN Topology

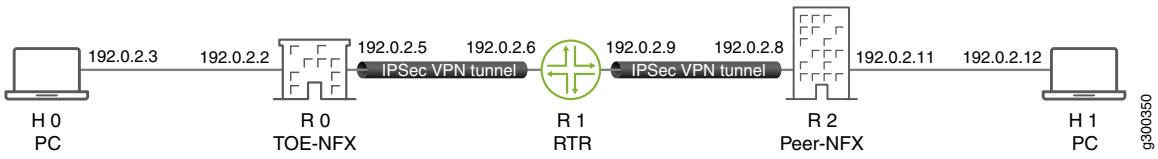


Table 5 on page 71 provides a complete list of the supported IKE protocols, tunnel modes, Phase 1 negotiation mode, authentication method or algorithm, encryption algorithm, DH groups supported for the IKE authentication and encryption (Phase1, IKE Proposal), and for IPsec authentication and encryption (Phase2, IPsec Proposal).

Table 5: VPN Combination Matrix

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	pre-shared-keys	sha-256	group14	3des-cbc
IKEv2			rsa-signatures-2048	sha-384	group19	aes-128-cbc
			ecdsa-signatures-256		group20	aes-128-gcm
			ecdsa-signatures-384		group24	aes-192-cbc
						aes-256-cbc
						aes-256-gcm

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha1-96	group14	ESP	3des-cbc
IKEv2			hmac-sha-256-128	group19		aes-128-cbc
				group20		aes-128-gcm
				group24		aes-192-cbc
						aes-192-gcm
						aes-256-cbc
						aes-256-gcm

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

Configuring an IPsec VPN with a Preshared Key for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using a preshared key as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 6 on page 72

Table 6: IKE or IPsec Authentication and Encryption

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

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

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

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

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

1. Configure the IKE proposal.

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

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

2. Configure the IKE policy.

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

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

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

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

> **NOTE:** The preshared key can alternatively be entered in hexadecimal format. For example:
>
> ```
> [edit]
> user@host# prompt security ike policy ike-policy1 hexadecimal
> New hexadecimal (secret):
> Retype new hexadecimal (secret) (secret):
> ```
>
> Here, the hexadecimal preshared key can be **cc2014bae9876543**.

3. Configure the IPsec proposal.

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

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

4. Configure the IPsec policy.

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

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

5. Configure the IKE.

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

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

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

6. Configure the VPN.

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

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

7. Configure the outbound flow policies.

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

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

8. Configure the inbound flow policies.

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

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

9. Commit your configuration.

```
user@host# commit
```

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

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

1. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method pre-shared-keys
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha256
user@host# set proposal ike-proposal1 encryption-algorithm 3des-cbc
```

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

2. Configure the IKE policy.

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

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

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

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

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

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

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

3. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set proposal ipsec-proposal1 encryption-algorithm 3des-cbc
```

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

4. Configure the IPsec policy.

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

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

5. Configure the IKE.

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

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

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

6. Configure the VPN.

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

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

7. Configure the outbound flow policies.

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

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

8. Configure the inbound flow policies.

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

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

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

9. Commit your configuration.

```
user@host# commit
```

Configuring an IPsec VPN with an RSA Signature for IKE Authentication

The following section provides an example to configure Junos OS devices for IPsec VPN using RSA Signature as IKE Authentication method, whereas, the algorithms used in IKE/IPsec authentication/encryption is as shown in the following table. In this section, you configure devices running Junos OS for IPsec VPN using an RSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 7 on page 79.

Table 7: IKE/IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE) Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	rsa-signatures-2048	sha-256	group14	aes-128-cbc

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

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

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

1. Configure the PKI. See Example: Configuring PKI.

2. Generate the RSA key pair. See Example: Generating a Public-Private Key Pair.

3. Generate and load the CA certificate. See Example: Loading CA and Local Certificates Manually.

4. Load the CRL. See Example: Manually Loading a CRL onto the Device .

5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.

6. Configure the IKE proposal.

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

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

7. Configure the IKE policy.

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

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

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
```

```
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set ipsec-proposal1 encryption-algorithm aes-128-cbc
```

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

9. Configure the IPsec policy.

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

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

10. Configure the IKE.

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

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

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

11. Configure VPN.

```
[edit security ipsec]
user@host# vpn vpn1 ike gateway gw1
user@host# vpn vpn1 ike ipsec-policy ipsec-policy1
```

```
user@host# vpn vpn1 bind-interface st0.0
```

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

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

12. Configure the outbound flow policies.

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

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

13. Configure the inbound flow policies.

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

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

14. Commit the configuration.

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

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

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

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

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

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

7. Configure the IKE policy.

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

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

8. Configure the IPsec proposal.

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

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

9. Configure the IPsec policy.

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

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

10. Configure IKE.

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

> **NOTE:** Here, **192.0.2.5** is the peer VPN endpoint IP, **192.0.2.8** is the local VPN endpoint IP, and **ge-0/0/2** is the local outbound interface as VPN endpoint. The following configuration is also needed for IKEv2.
>
> ```
> [edit security ike]
> user@host# set gw1 version v2-only
> ```

11. Configure VPN.

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

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

```
user@host# set routing-options static route 192.0.2.1/24 qualified-next-hop st0.0 preference 1
```

12. Configure the outbound flow policies.

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

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

13. Configure the inbound flow policies.

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

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

14. Commit the configuration.

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

Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using an ECDSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption are shown in Table 8 on page 86.

Table 8: IKE or IPsec Authentication and Encryption

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

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

Configuring IPsec VPN with ECDSA signature IKE authentication on the Initiator

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

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

6. Configure the IKE proposal.

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

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

7. Configure the IKE policy.

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

8. Configure the IPsec proposal.

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

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

9. Configure the IPsec policy.

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

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

10. Configure IKE.

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

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

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

11. Configure the VPN.

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

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

12. Configure the outbound flow policies.

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

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

13. Configure the inbound flow policies.

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

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

14. Commit your configuration.

```
user@host# commit
```

Configuring IPsec VPN with ECDSA signature IKE authentication on the Responder

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

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

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method ecdsa-signatures-256
```

```
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha-384
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

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

6. Configure the IKE policy.

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

7. Configure the IPsec proposal.

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

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

8. Configure the IPsec policy.

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

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

9. Configure the IKE.

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

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

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

10. Configure the VPN.

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

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

11. Configure the outbound flow policies.

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

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

12. Configure the inbound flow policies.

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

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

13. Commit your configuration.

```
user@host# commit
```

RELATED DOCUMENTATION

- Public Key Infrastructure Feature Guide for Security Devices

8

CHAPTER

Configuring Security Flow Policies

Understanding a Security Flow Policy on a Device Running Junos OS | 94

Understanding a Security Flow Policy on a Device Running Junos OS

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

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

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

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

- Configuring a Security Flow Policy in Firewall Bypass Mode on page 94
- Configuring a Security Policy in Firewall Discard Mode on page 95
- Configuring a Security Flow Policy in IPsec Protect Mode on page 95

Configuring a Security Flow Policy in Firewall Bypass Mode

To configure a security flow policy for firewall bypass mode:

- Configure the security policies.

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

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

Configuring a Security Policy in Firewall Discard Mode

To configure a security flow policy for firewall discard mode:

- Configure the security policies.

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

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

Configuring a Security Flow Policy in IPsec Protect Mode

To configure a security flow policy for IPSec protect mode:

1. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
```

```
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 198.51.100.14/24 qualified-next-hop st0.0 preference 1
```

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

2. Configure the security policies.

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

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

CHAPTER 9

Configuring Traffic Filtering Rules

Understanding Protocol Support | **98**

Configuring Traffic Filter Rules | **99**

Configuring Default Deny-All and Reject Rules | **100**

Logging the Dropped Packets Using Default Deny-all Option | **101**

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets | **102**

Configuring Default Reject Rules for Source Address Spoofing | **103**

Configuring Default Reject Rules with IP Options | **103**

Configuring Default Reject Rules | **105**

Configuring the Device to Drop Unassigned IPv6 Packets | **106**

Understanding Protocol Support

You can configure the devices running Junos OS to perform stateful network traffic filtering on network packets using network traffic protocols and network fields as described in Table 9 on page 98.

Table 9: Network Traffic Protocols and Fields

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

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

- IPsec
- IKE
- SSH

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

- OSPF
- BGP
- RIP

Configuring Traffic Filter Rules

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

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

1. Configure a zone and its interfaces.

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

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

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

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

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

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

Configuring Default Deny-All and Reject Rules

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

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

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

- Invalid fragments
- Fragmented IP packets that cannot be reassembled completely
- Where the source address is equal to the address of the network interface
- Where the source address does not belong to the networks associated with the network interface
- Where the source address is defined as being on a broadcast network
- Where the source address is defined as being on a multicast network
- Where the source address is defined as being a loopback address
- Where the source address is a multicast packet
- Where the source or destination address is a link-local address
- Where the source or destination address is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4
- Where the source or destination address is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6
- With the IP option Loose Source Routing, Strict Source Routing, or Record Route is specified

Logging the Dropped Packets Using Default Deny-all Option

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

Before you begin, log in with your root account and edit the configuration.

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

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

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

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

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

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

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

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

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

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

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

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets

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

Before you begin, log in with your root account and edit the configuration.

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

To configure mandatory reject rules:

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

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

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

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

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

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

Configuring Default Reject Rules for Source Address Spoofing

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

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

Before you begin, log in with your root account and edit the configuration.

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

To configure default reject rules to log source address spoofing:

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

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

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

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

Configuring Default Reject Rules with IP Options

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

Before you begin, log in with your root account.

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

To configure the default reject rules with IP options:

1. Configure the screen features to enable IP options.

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

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

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

Configuring Default Reject Rules

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

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

Before you begin, log in with your root account and edit the configuration.

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

To configure default reject rules:

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

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

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

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

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

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

4. Configure the mandatory TCP reject rule.

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

Configuring the Device to Drop Unassigned IPv6 Packets

Before you configure the device to drop unassigned IPv6 packets, check the default configuration status of the device. From the operational mode enter the **show usp flow** configuration command.

In the output, the advanced options **no_drop_unassigned_ipv6_address: disabled (default)** option indicates that by default the device drops unassigned IPv6 packets.

To enable the device to drop unassigned IPv6 packets, use the following command:

```
user@host# set security flow advanced-options no-drop-unassigned-ipv6-address
user@host# commit
```

To enable users to revert to the default configuration, use the following command:

```
user@host# delete security flow advanced-options no-drop-unassigned-ipv6-address
user@host# commit
```

10
CHAPTER

Configuring Network Attacks

Configuring IP Teardrop Attack Screen | **109**

Configuring TCP Land Attack Screen | **110**

Configuring ICMP Fragment Screen | **112**

Configuring Ping-Of-Death Attack Screen | **113**

Configuring tcp-no-flag Attack Screen | **115**

Configuring TCP SYN-FIN Attack Screen | **116**

Configuring TCP fin-no-ack Attack Screen | **118**

Configuring UDP Bomb Attack Screen | **119**

Configuring UDP CHARGEN DoS Attack Screen | **119**

Configuring TCP SYN and RST Attack Screen | **121**

Configuring ICMP Flood Attack Screen | **123**

Configuring TCP SYN Flood Attack Screen | **124**

Configuring TCP Port Scan Attack Screen | **126**

Configuring UDP Port Scan Attack Screen | **127**

Configuring IP Sweep Attack Screen | **129**

Configuring IP Teardrop Attack Screen

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

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

To enable detection of a teardrop attack:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen ip tear-drop
```

```
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP Land Attack Screen

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

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

To enable detection of a TCP land attack:

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

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

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

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

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

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

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

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

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

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring ICMP Fragment Screen

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

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

To enable detection of an ICMP fragment IDS attack:

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

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

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

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

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

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

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

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

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

RELATED DOCUMENTATION

IDP Extended Package Configuration Overview

Configuring Ping-Of-Death Attack Screen

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

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

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

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

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

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

```
user@host# set security zones security-zone trustZone host-inbound-traffic system-services all
```

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

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

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

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

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

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring tcp-no-flag Attack Screen

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

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

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

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen tcp tcp-no-flag
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP SYN-FIN Attack Screen

This topic describes how to configure detection of a TCP SYN-FIN attack.

A TCP header with the SYN and FIN flags set is anomalous TCP behavior causing various responses from the recipient, depending on the OS. Blocking packets with SYN and FIN flags helps prevent the OS system probes.

To enable detection of TCP SYN-FIN bits:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen tcp syn-fin
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP fin-no-ack Attack Screen

This topic describes how to configure detection of TCP **fin-no-ack** attack. A TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior.

To enable detection of FIN bits with no ACK bit IDS option:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen tcp fin-no-ack
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring UDP Bomb Attack Screen

If the UDP length specified is less than the IP length specified then the malformed packet type is associated with a denial-of-service attempt. By default, NFX drops these packets. No configuration is required.

Configuring UDP CHARGEN DoS Attack Screen

This topic describes how to configure protection from a UDP CHARGEN DoS attack.

NOTE: UDP packet is detected with a source port of 7 and a destination port of 19 is an attack.

To enable detection of a UDP CHARGEN DoS attack:

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

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

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

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

3. Configure security policies from **untrustZone** to the **trustZone** with the Junos OS predefined application **junos-chargen**.

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

4. Configure syslog.

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

5. To allow the packet to reach the destination, change the policy configuration from **deny** to **permit**.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP SYN and RST Attack Screen

This topic describes how to configure TCP packet when the SYN and RST flags are set.

To enable detection of a TCP SYN and RST attack:

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

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

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

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

3. Configure the IDP custom-attack signatures.

```
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match from-zone any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match source-address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match to-zone any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match destination-address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match application default
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match attacks custom-attacks syn_rst
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then action no-action
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then notification log-attacks
user@host# set security idp active-policy idpengine
user@host# set security idp custom-attack syn_rst severity info
user@host# set security idp custom-attack syn_rst attack-type signature context packet
user@host# set security idp custom-attack syn_rst attack-type signature pattern
user@host# set security idp custom-attack syn_rst attack-type signature direction any
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp tcp-flags rst
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp tcp-flags syn
```

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

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

5. Configure security **tcp-session** option in flow.

```
user@host# set security flow tcp-session no-syn-check
user@host# set security flow tcp-session no-sequence-check
```

6. Configure syslog.

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

7. To allow the traffic to reach the destination, configure the **tcp-session** option.

```
user@host# set security flow tcp-session relax-check
```

8. Commit the configuration.

```
user@host# commit
```

Configuring ICMP Flood Attack Screen

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

An ICMP flood typically occurs when an ICMP echo request overloads the victim with many requests such that the ICMP echo request spends all its resources responding until it can no longer process valid network traffic. When enabling the ICMP flood protection feature, you can set a threshold that, once exceeded, invokes the ICMP flood attack protection feature.

To enable detection of an ICMP flood attack:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen icmp flood
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

```
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP SYN Flood Attack Screen

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

A SYN flood occurs when a host is so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

To enable detection of a TCP SYN flood attack:

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

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen tcp syn-flood
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring TCP Port Scan Attack Screen

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

A port scan occurs when one source IP address sends an IP packet containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval.

To enable detection of a TCP port scan attack:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen tcp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring UDP Port Scan Attack Screen

This topic describes how to configure detection of a UDP port scan attack.

These attacks scan the target IP addresses for open, listening, or responsive services by targeting multiple protocols or ports on one or more target IP address using obvious (sequentially numbered) patterns of the target protocol or port numbers. The patterns are derived by randomizing the protocol or port numbers and randomizing the time delays between the transmissions.

To enable detection of a UDP port scan attack:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen udp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

Configuring IP Sweep Attack Screen

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

An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined time interval (5000 microseconds is the default value). The purpose of this attack is to send ICMP packets—typically echo requests—to various hosts in the hope that at least one replies, thus uncovering an address to target.

To enable detection of an IP sweep attack:

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

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

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

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

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

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

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

```
user@host# set security screen ids-option untrustScreen icmp ip-sweep
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

```
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

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

6. Commit the configuration.

```
user@host# commit
```

11

CHAPTER

Configuring the IDP Extended Package

IDP Extended Package Configuration Overview | 132

IDP Extended Package Configuration Overview

The Junos OS Intrusion Detection and Prevention (IDP) policy enables you to selectively enforce various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. It allows you to define policy rules to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy defines how your device handles the network traffic. It allows you to enforce various attack detection and prevention techniques on traffic traversing your network.

A policy is made up of rule bases, and each rule base contains a set of rules. You define rule parameters, such as traffic match conditions, action, and logging requirements, then add the rules to rule bases. After you create an IDP policy by adding rules in one or more rule bases, you can select that policy to be the active policy on your device.

To configure the IDP extended package (IPS-EP) perform the following steps:

1. Enable IPS in a security policy. See *Configuring IDP Policy Rules and IDP Rulebases* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

2. Configure IDP policy rules, IDP rule bases, and IDP rule actions. See *Configuring IDP Policy Rules and IDP Rulebases* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

3. Configure IDP custom signatures. See *Understanding IDP Signature-Based Attacks* and *Example: Configuring IDP Signature-Based Attacks* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

4. Update the IDP signature database. See *Updating the IDP Signature Database Overview* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

12
CHAPTER

Performing Self-Tests on a Device

Understanding FIPS Self-Tests | **134**

Example: Configuring FIPS Self-Tests | **134**

Verifying That FIPS Self-Tests Are Taking Place | **136**

Understanding FIPS Self-Tests

The cryptographic module enforces security rules to ensure that an NFX150 device running 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 system 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
- **ssh_ipsec_kats**—KAT for SSH IPsec Toolkit cryptographic implementation

The KAT self-tests are performed automatically at startup and reboot, regardless of whether FIPS mode is enabled on the NFX150 device. Conditional self-tests are also performed automatically to verify digitally signed software packages, generated random numbers, RSA and DSA key pairs, and manually entered keys.

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

If the system fails a KAT, it writes the details to a system log file, enters FIPS error state (panic), and reboots the NFX150 device.

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

Example: Configuring FIPS Self-Tests

IN THIS SECTION

- Hardware and Software Requirements | **135**
- Overview | **135**
- Configuration | **135**

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
- **openssl_kats**—KAT for OpenSSL cryptographic implementation
- **ssh_ipsec_kats**—KAT for SSH IPsec Toolkit 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

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then 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:

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

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

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

```
[edit system fips self-test]
user@host# 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.

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

Verifying That FIPS Self-Tests Are Taking Place

Purpose

Verify that FIPS self-tests are taking place on the NFX150 device.

Action

You can run FIPS self-tests manually by issuing the **request system fips self-test** command.

```
{master:0}

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

```
Testing kernel KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-384 Known Answer Test:                     Passed
  SHA-2-512 Known Answer Test:                     Passed
  AES128-CMAC Known Answer Test:                   Passed
  AES-CBC Known Answer Test:                       Passed
Testing MacSec KATS:
  AES128-CMAC Known Answer Test:                   Passed
  AES256-CMAC Known Answer Test:                   Passed
  AES-KEYWRAP Known Answer Test:                   Passed
Testing libmd KATS:
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-512 Known Answer Test:                     Passed
Testing OpenSSL KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  FIPS ECDSA Known Answer Test:                    Passed
  FIPS ECDH Known Answer Test:                     Passed
  FIPS RSA Known Answer Test:                      Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  ECDSA-SIGN Known Answer Test:                    Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
  KDF-SSH-SHA256 Known Answer Test:                Passed
  KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test:    Passed
  KAS-FFC-EPHEM-NOKC Known Answer Test:            Passed
Testing QuickSec 7.0 KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
```

```
  AES-GCM Known Answer Test:                        Passed
  SSH-RSA-ENC Known Answer Test:                    Passed
  SSH-RSA-SIGN Known Answer Test:                   Passed
  SSH-ECDSA-SIGN Known Answer Test:                 Passed
  KDF-IKE-V1 Known Answer Test:                     Passed
  KDF-IKE-V2 Known Answer Test:                     Passed
Testing QuickSec KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:          Passed
  DES3-CBC Known Answer Test:                       Passed
  HMAC-SHA1 Known Answer Test:                      Passed
  HMAC-SHA2-224 Known Answer Test:                  Passed
  HMAC-SHA2-256 Known Answer Test:                  Passed
  HMAC-SHA2-384 Known Answer Test:                  Passed
  HMAC-SHA2-512 Known Answer Test:                  Passed
  AES-CBC Known Answer Test:                        Passed
  AES-GCM Known Answer Test:                        Passed
  SSH-RSA-ENC Known Answer Test:                    Passed
  SSH-RSA-SIGN Known Answer Test:                   Passed
  KDF-IKE-V1 Known Answer Test:                     Passed
  KDF-IKE-V2 Known Answer Test:                     Passed
Testing SSH IPsec KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:          Passed
  DES3-CBC Known Answer Test:                       Passed
  HMAC-SHA1 Known Answer Test:                      Passed
  HMAC-SHA2-256 Known Answer Test:                  Passed
  AES-CBC Known Answer Test:                        Passed
  SSH-RSA-ENC Known Answer Test:                    Passed
  SSH-RSA-SIGN Known Answer Test:                   Passed
  KDF-IKE-V1 Known Answer Test:                     Passed
Testing file integrity:
  File integrity Known Answer Test:                 Passed
Testing crypto integrity:
  Crypto integrity Known Answer Test:               Passed
Expect an exec Authentication error...
/sbin/kats/run-tests: /sbin/kats/cannot-exec: Authentication error
```

After a self-test is run on the NFX150 device, the system log (syslog) file is updated to display the known answer tests (KATs) that are executed. To view the system log file, issue the command **file show /var/log/messages**. The system log file displays the date and time at which each KAT was executed, the name of the test, and its status.

RELATED DOCUMENTATION

13

CHAPTER

Junos-FIPS Configuration Restrictions

Understanding Remote Access for Junos OS in FIPS Mode | **141**

Understanding Configuration Limitations and Restrictions on Junos OS in FIPS Mode | **141**

Understanding Remote Access for Junos OS in FIPS Mode

When the NFX150 device is in Junos OS in FIPS mode, only SSH is available as a remote access service. To secure the information sent on administrative connections, use SSHv2 for CLI configuration. For SSH configuration information, see the *Junos OS System Basics Configuration Guide.*

> **BEST PRACTICE:** For FIPS compliance, configure the NFX150 device over SSH connections because they are encrypted connections.

The Ethernet management (MGMT) port on the NFX150 device is enabled by default.

In Junos OS in FIPS mode, all critical security parameters (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. However, as the Crypto Officer, you can enter user authentication data in plain text.

RELATED DOCUMENTATION

Understanding Configuration Limitations and Restrictions on Junos OS in FIPS Mode | 141

Junos OS System Basics and Services Command Reference

Understanding Configuration Limitations and Restrictions on Junos OS in FIPS Mode

In FIPS mode, an NFX150 device operates as a non-modifiable operational environment in which only files shipped as part of Junos OS can be executed.

In contrast to non-FIPS mode, Junos OS in FIPS mode:

- Conforms to FIPS 140-2.
- Requires special installation procedures.
- Mandates the use of internal, manual IPsec tunnels with specific requirements.
- Limits services used for remote access.

- Allows only the use of approved ciphers.
- Requires user logout on disconnect at the console.
- Sets strict requirements for passwords.
- Requires special system logging considerations.
- Disables the following Junos OS protocols and services so that you cannot configure them. Attempts to configure these services or to load configurations with these services configured result in a configuration syntax error.
  - finger
  - FTP
  - rlogin
  - rsh
  - Telnet
  - Trivial File Transfer Protocol (TFTP)
  - Transport Layer Security (TLS) protocol
  - xnm-clear-text

  If you try to load a configuration that includes statements not supported by Junos OS in FIPS mode, you see a warning message. For example, suppose you attempt to configure Telnet for remote access:

```
[edit]
crypto-officer:fips# set system services telnet
```

  You receive the following warning and cannot add the **system services telnet** statement to the loaded configuration:

```
[edit]
'telnet'
warning: not allowed in JUNOS-FIPS; ignored
```

RELATED DOCUMENTATION

Understanding Junos OS in FIPS Mode | 14

Understanding Remote Access for Junos OS in FIPS Mode | 141

Understanding FIPS Terminology and Supported Cryptographic Algorithms | 15

Understanding Password Specifications and Guidelines for Junos OS in FIPS Mode | 30

14

CHAPTER

Configuration Statements

fips (FIPS) | **145**

level (FIPS) | **146**

chassis level (FIPS) | **147**

fips (FIPS)

Syntax

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

Hierarchy Level

```
[edit system]
```

Description

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

The remaining statements are explained separately.

Required Privilege Level

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

level (FIPS)

Syntax

```
level level;
```

Hierarchy Level

```
[edit system fips]
```

Release Information

Statement introduced in Junos OS Release 18.2R1 for NFX Series.

Description

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

Compared to non-FIPS mode, Junos OS in FIPS mode is a nonmodifiable operational environment with limitations. (See "Understanding Configuration Limitations and Restrictions on Junos OS in FIPS Mode" on page 141.)

Options

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

Range: 0 through 4

> **NOTE:** To enable Junos OS in FIPS mode on NFX150 device , set **level** to 1. Only level 1 is supported on the device.

Required Privilege Level

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

chassis level (FIPS)

Syntax

```
level level;
```

Hierarchy Level

```
[edit system fips chassis]
```

Release Information

Statement introduced in Junos OS Release 18.2R1 for NFX150 Series.

Description

Set the chassis level to configure chassis boundary fips.

Options

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

Range: 0 through 4

Required Privilege Level

maintenance—To add and view this statement in the configuration.

15
CHAPTER

Operational Commands

request system zeroize (FIPS) | **149**

request system zeroize (FIPS)

Syntax

request system zeroize

Description

Remove all configuration information on the Routing Engines hypervisor and reset all key values. 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

RELATED DOCUMENTATION

Understanding Zeroization to Clear System Data for FIPS Mode | 33

Zeroizing the System | 35

List of Sample Output

request system zeroize (FIPS) on page 149

Output Fields

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

Sample Output

request system zeroize (FIPS)

```
{master:0}

root@device: fips> request system zeroize

warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (no) yes

warning: zeroizing re0
: Switching to runlevel: 6
```

```
INIT: Sending processes the TERM signal

root@porter3s1E-p2a-02:fips> stopping rsyslogd ... done
Stopping OpenBSD Secure Shell server: sshdno /usr/sbin/sshd found; none killed
jdm
Error response from daemon: Driver aufs failed to remove root filesystem
31ff0cdaaa47d367954de0ca657bcf2d5e1913be3bd90f3d12ed32dc266c6819: rename
/var/lib/docker/aufs/mnt/31ff0cdaaa47d367954de0ca657bcf2d5e1913be3bd90f3d12ed32dc266c6819

/var/lib/docker/aufs/mnt/31ff0cdaaa47d367954de0ca657bcf2d5e1913be3bd90f3d12ed32dc266c6819-removing:
 device or resource busy
Error: failed to remove containers: [jdm]
[  OK  ]
Stopping atd: OK
Unmounting cgroups...umount: /sys/fs/cgroup: target is busy
        (In some cases useful info about processes that
         use the device is found by lsof(8) or fuser(1).)
Done
Stopping system message bus: dbus.
stopping DNS forwarder and DHCP server: dnsmasq... stopped /usr/bin/dnsmasq (pid
11215 11213)
done.
Stopping docker: /etc/init.d/functions: line 286: usleep: command not found
[  OK  ]
Unmounting fuse control filesystem.
Unloading fuse module failed!
Shutting down irqbalance: stopped irqbalance (pid 3357)
done
Stopping ntpd: done
stopping rsyslogd ... done
Stopping internet superserver: xinetd.

Waiting for sanlock to stop: Success

Clearing ebtables rulesets: filter nat broute done. ok
Kdump has been stopped.
Stopping crond: OK
Stopping S.M.A.R.T. daemon: smartd.
Stopping fan control daemon: fancontrol... no process in pidfile
'/var/run/fancontrol.pid' found; none killed
done.
Stopping sensors logging daemon: sensord... stopped /usr/sbin/sensord (pid 3738)
done.
 * Stopping virtualization library daemon: libvirtd
```

```
Deconfiguring network interfaces... done.
Stopping tcsd: tcsd (pid 3826
5058) is running...
/etc/init.d/functions: line 286: usleep: command not found
OK
Stopping redis-server...
/etc/rc6.d/K99lte.init: line 38: ltelog: command not found
cp: cannot stat '/var/platform/lte_vm_xml_params': No such file or directory
/
error: failed to connect to the hypervisor
error: no valid connection
error: Failed to connect socket to '/var/run/libvirt/libvirt-sock': No such file
or directory

Sending all processes the TERM signal...
Merlin_daemon[2993]: Exiting daemon
Sending all processes the KILL signal...
Unmounting remote filesystems...
Deactivating swap...
Unmounting local filesystems...
```