Juniper Networks, Inc.

FIPS 140-3 Non-Proprietary Security Policy

Juniper Networks EX4100

Version: Junos OS 22.4R2

Prepared for:

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408.745.2000
1.888 JUNIPER
www.juniper.net

Prepared by:

TERON LABS
www.teronlabs.com

Table of Contents

1 General 5

1.1 Overview 5

1.2 Security Levels 5

2 Cryptographic Module Specification 5

2.1 Description 5

2.2 Tested and Vendor Affirmed Module Version and Identification 7

2.3 Excluded Components 8

2.4 Modes of Operation 8

2.5 Algorithms 8

2.6 Security Function Implementations 10

2.7 Algorithm Specific Information 12

2.8 RBG and Entropy 13

2.9 Key Generation 13

2.10 Key Establishment 13

2.11 Industry Protocols 13

3 Cryptographic Module Interfaces 14

3.1 Ports and Interfaces 14

4 Roles, Services, and Authentication 14

4.1 Authentication Methods 14

4.2 Roles 15

4.3 Approved Services 15

4.4 Non-Approved Services 18

4.5 External Software/Firmware Loaded 18

5 Software/Firmware Security 18

5.1 Integrity Techniques 18

5.2 Initiate on Demand 18

6 Operational Environment 18

6.1 Operational Environment Type and Requirements 18

6.2 Configuration Settings and Restrictions 19

7 Physical Security 19

8 Non-Invasive Security 19

9 Sensitive Security Parameters Management 19

9.1 Storage Areas 19

9.2 SSP Input-Output Methods 19

9.3 SSP Zeroization Methods 20
9.4 SSPs 20
9.5 Transitions 23
10 Self-Tests 23
10.1 Pre-Operational Self-Tests 24
10.2 Conditional Self-Tests 24
10.3 Periodic Self-Test Information 25
10.4 Error States 26
10.5 Operator Initiation of Self-Tests 27
11 Life-Cycle Assurance 27
11.1 Installation, Initialization, and Startup Procedures 27
11.2 Administrator Guidance 27
11.2.1 Installing the Junos OS firmware image 27
11.2.2 Configure the device for the Approved mode 28
11.2.3 Zeroizing the System 28
11.3 Non-Administrator Guidance 29
11.4 Design and Rules 29
11.4.1 Module Design Rules 29
11.4.2 Module Operation Rules 29
11.5 Maintenance Requirements 29
11.6 End of Life 29
12 Mitigation of Other Attacks 30

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Hardware 7
Table 3: Modes List and Description 8
Table 4: Approved Algorithms - OpenSSL 1.0.2 9
Table 5: Approved Algorithms - OpenSSL 1.1.1 9
Table 6: Approved Algorithms - Kernel 10
Table 7: Approved Algorithms - LibMD 10
Table 8: Vendor-Affirmed Algorithms 10
Table 9: Security Function Implementations 12
Table 10: Entropy Certificates 13
Table 11: Entropy Sources 13
Table 12: Ports and Interfaces 14
Table 13: Authentication Methods 15
Table 14: Roles 15
Table 15: Approved Services 18
Table 16: Mechanisms and Actions Required 19
Table 17: Storage Areas 19
Table 18: SSP Input-Output Methods 20
Table 19: SSP Zeroization Methods 20
Table 20: SSP Table 1 22
Table 21: SSP Table 2 23
Table 22: Pre-Operational Self-Tests 24
Table 23: Conditional Self-Tests 25
Table 24: Pre-Operational Periodic Information 25
Table 25: Conditional Periodic Information 26
Table 26: Error States 27

List of Figures

Figure 1 – EX4100-F-48P Ethernet Switch (front) 6
Figure 2 – EX4100-F-48P Ethernet Switch (rear) 6
Figure 3 – EX4100-F-48T Ethernet Switch (front) 6
Figure 4 – EX4100-F-48T Ethernet Switch (rear) 6
Figure 5 – EX4100-F-24P Ethernet Switch (front) 7
Figure 6 – EX4100-F-24P Ethernet Switch (rear) 7
Figure 7 – EX4100-F-24T Ethernet Switch (front) 7
Figure 8 – EX4100-F-24T Ethernet Switch (rear) 7

1 General

1.1 Overview

This is a non-proprietary Cryptographic Module Security Policy for the Juniper Networks EX4100-F-48P, EX4100-F-48T, EX4100-F-24P, EX4100-F-24T Ethernet Switches, hereafter referred to as the cryptographic module.

1.2 Security Levels

The cryptographic module is designed to meet FIPS 140-3 Level 1 overall. The table below shows the security levels claimed for each section of the security requirements.

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	2
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	1
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Juniper Networks EX4100 line of Ethernet Switches offers a secure, cloud-ready portfolio of access switches ideal for enterprise branch, campus, and data center networks.

This FIPS 140-3 validation comprises the following EX series switch models EX4100-F-48P, EX4100-F-48T, EX4100-F-24P, and EX4100-F-24T.

The cryptographic module runs Junos OS, Juniper's reliable, high-performance, modular network operating system that is supported across all of Juniper's physical and virtual routing, switching, and security platforms.

The cryptographic module provides for an encrypted connection, using SSH, between the management station and the module. All other data input or output from the modules are considered plaintext for this FIPS 140-3 validation.

Module Type:
The cryptographic module is a Hardware cryptographic module.

Module Embodiment:
The cryptographic module is defined as a MultiChipStand module that executes Junos OS 22.4R2 firmware on any of the identified Juniper Networks devices.

Module Characteristics:
There are no additional characteristics relevant to this module.

Cryptographic Boundary:

The Tested Operational Environment Physical Perimeter (TOEPP) is defined as the outer edge of the chassis. The chassis is a rigid sheet-metal structure that houses all components of the device. The cryptographic boundary encompasses the entire TOEPP.

The cryptographic module is FIPS-compliant when installed and configured with Junos OS 22.4R2 validated firmware as specified in section 11.1.

The physical form of the module is depicted in Figures 1 to 8.

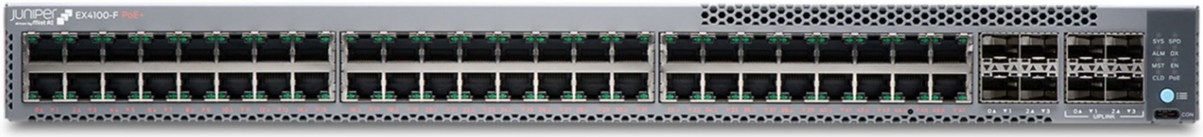

Figure 1 – EX4100-F-48P Ethernet Switch (front)

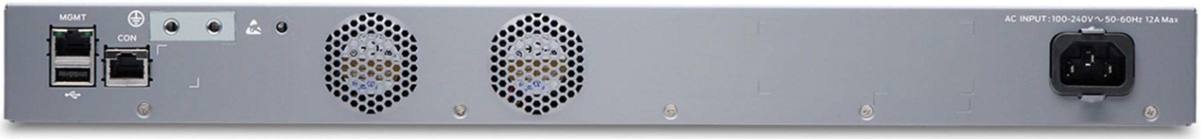

Figure 2 – EX4100-F-48P Ethernet Switch (rear)

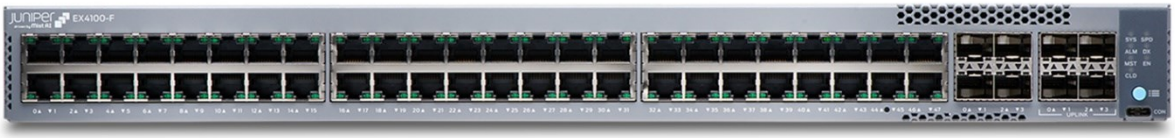

Figure 3 – EX4100-F-48T Ethernet Switch (front)

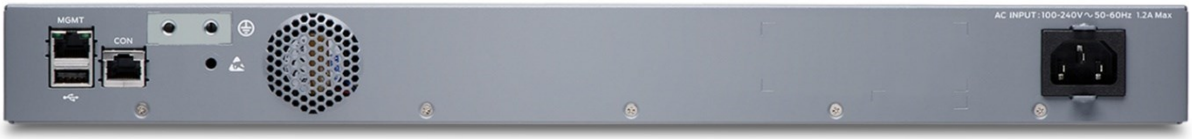

Figure 4 – EX4100-F-48T Ethernet Switch (rear)

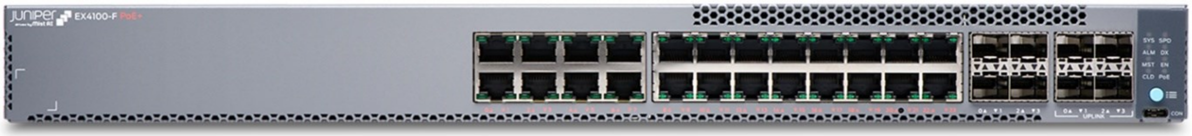

Figure 5 – EX4100-F-24P Ethernet Switch (front)

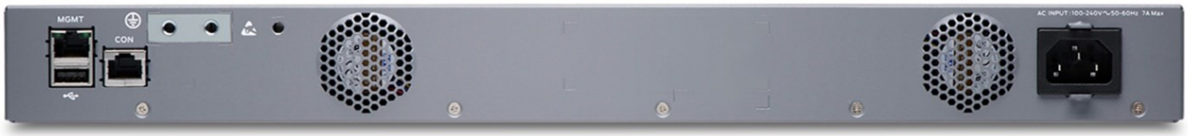

Figure 6 – EX4100-F-24P Ethernet Switch (rear)

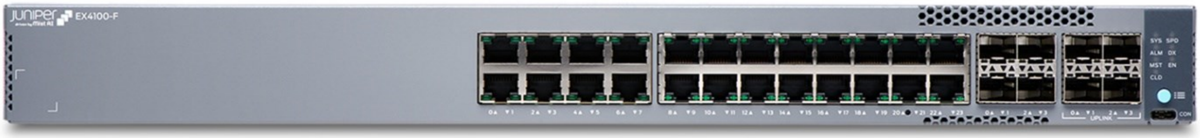

Figure 7 – EX4100-F-24T Ethernet Switch (front)

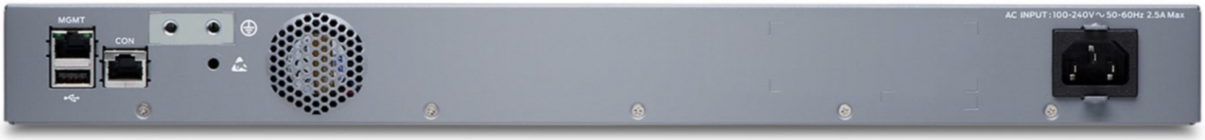

Figure 8 – EX4100-F-24T Ethernet Switch (rear)

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

The following models of the module were tested.

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
EX4100-F-48P	EX4100-F-48P	Junos OS 22.4R2.8	ARM-cortex A72 64-bit, single core	48 x 10 MB/100 MB/1GbE PoE+ access ports
EX4100-F-48T	EX4100-F-48T	Junos OS 22.4R2.8	ARM-cortex A72 64-bit, single core	48 x 10 MB/100 MB/1GbE access ports
EX4100-F-24P	EX4100-F-24P	Junos OS 22.4R2.8	ARM-cortex A72 64-bit, single core	24 x 10 MB/100 MB/1GbE PoE+ access ports
EX4100-F-24T	EX4100-F-24T	Junos OS 22.4R2.8	ARM-cortex A72 64-bit, single core	24 x 10 MB/100 MB/1GbE access ports

Table 2: Tested Module Identification – Hardware

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets): N/A

The module is not classified as software, firmware, or hybrid; thus, this section is not applicable.

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware: N/A

The module is not classified as hybrid disjoint hardware; thus, this section is not applicable.

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid: N/A

The module is not classified as software, firmware, or hybrid; thus, this section is not applicable.
N/A for this module.

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid: N/A

There are no vendor-affirmed operational environments claimed.

N/A for this module.

2.3 Excluded Components

No components are excluded from the requirements of FIPS PUB 140-3.

2.4 Modes of Operation

The module supports an Approved mode only. The module enters Approved mode as a result of successful installation, initialization and configuration steps described in section 11. Until these procedures have been followed, the module is non-compliant.

Mode Name	Description	Type	Status Indicator
Approved	Approved mode of operation.	Approved	Suffix string ":fips" in the cli prompt

Table 3: Modes List and Description

2.5 Algorithms

Approved Algorithms:

Although the module may have been tested for additional algorithms or modes, only those listed below are utilized by the module.

OpenSSL 1.0.2

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4301	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4301	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
ECDSA KeyGen (FIPS186-4)	A4301	Curve - P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4301	Curve - P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4301	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4301	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
HMAC-SHA-1	A4301	Key Length - Key Length: 160	FIPS 198-1
HMAC-SHA2-256	A4301	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-512	A4301	Key Length - Key Length: 512	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4301	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF SSH (CVL)	A4301	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A4301	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A4301	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A4301	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
SHA-1	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms - OpenSSL 1.0.2

OpenSSL 1.1.1

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-4)	A4302	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
SHA2-256	A4302	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 5: Approved Algorithms - OpenSSL 1.1.1

Kernel

Algorithm	CAVP Cert	Properties	Reference
HMAC DRBG	A4303	Prediction Resistance - Yes Mode - SHA2-256	SP 800-90A Rev. 1
HMAC-SHA2-256	A4303	Key Length - Key Length: 256	FIPS 198-1
SHA2-256	A4303	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-512	A4303	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4

Table 6: Approved Algorithms - Kernel

LibMD

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A4306	Key Length - Key Length: 112, 160	FIPS 198-1
HMAC-SHA2-256	A4306	Key Length - Key Length: 160, 256	FIPS 198-1
SHA-1	A4306	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-256	A4306	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-512	A4306	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 7: Approved Algorithms - LibMD

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key type:Asymmetric	N/A	SP 800-133r2 Section 4, example 1 direct output from DRBG.

Table 8: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

The module implements the security functions listed in the following table.

Name	Type	Description	Properties	Algorithms
Enc/Dec (SSH)	BC-UnAuth	Unauthenticated encryption for SSH		AES-CBC: (A4301) AES-CTR: (A4301)
KAS-SSC (SSH)	KAS-SSC	Key Agreement Scheme Shared Secret Computation for SSH		KAS-ECC-SSC Sp800-56Ar3: (A4301)

Name	Type	Description	Properties	Algorithms
KeyGen (SSH)	AsymKeyPair-KeyGen CKG	Key Generation used for SSH authentication keys		ECDSA KeyGen (FIPS186-4): (A4301) ECDSA KeyVer (FIPS186-4): (A4301) RSA KeyGen (FIPS186-5): (A4301) HMAC DRBG: (A4303) CKG: ()
SigGen (SSH)	DigSig-SigGen	Signature Generation for peer authentication in SSH		HMAC DRBG: (A4303) ECDSA SigGen (FIPS186-4): (A4301) RSA SigGen (FIPS186-5): (A4301) SHA2-256: (A4301) SHA2-384: (A4301) SHA2-512: (A4301)
SigVer (SSH)	DigSig-SigVer	Signature Verification for peer authentication in SSH		ECDSA SigVer (FIPS186-4): (A4301) RSA SigVer (FIPS186-5): (A4301) SHA2-256: (A4301) SHA2-384: (A4301) SHA2-512: (A4301)
MAC (SSH)	MAC	Message authentication for SSH		HMAC-SHA-1: (A4301) HMAC-SHA2-256: (A4301) HMAC-SHA2-512: (A4301)
KAS KeyGen (SSH)	CKG KAS-KeyGen	Key Generation for Key Agreement in SSH		ECDSA KeyGen (FIPS186-4): (A4301) ECDSA KeyVer (FIPS186-4): (A4301) CKG: () HMAC DRBG: (A4303)
KDF (SSH)	KAS-135KDF	Key derivation function for SSH		KDF SSH: (A4301) SHA-1: (A4301) SHA2-256: (A4301) SHA2-384: (A4301) SHA2-512: (A4301)
Full KAS (SSH)	KAS-Full	Full Key Agreement for SSH	IG:IG D.F Scenario 2 path (2), split. Key confirmation:No Key derivation:KDF SSH (separately tested).	ECDSA KeyGen (FIPS186-4): (A4301) ECDSA KeyVer (FIPS186-4): (A4301) KAS-ECC-SSC Sp800-56Ar3: (A4301) SHA-1: (A4301) SHA2-256: (A4301) SHA2-384: (A4301) SHA2-512: (A4301) KDF SSH: (A4301)

Name	Type	Description	Properties	Algorithms
KTS (SSH)	KTS-Wrap KTS-Unwrap	Key transport using SSH as per IG D.G provisions	Standard:SP 800-38F IG D.G:Approved key wrapping key using combination (encryption + authentication) method. Caveat:Key establishment methodology provides between 112 and 256 bits of security strength	AES-CBC: (A4301) AES-CTR: (A4301) HMAC-SHA-1: (A4301) HMAC-SHA2-256: (A4301) HMAC-SHA2-512: (A4301)
SHA (LibMD)	SHA	Message Digest Generation		SHA-1: (A4306) SHA2-256: (A4306) SHA2-512: (A4306)
MAC (LibMD)	MAC	Message Authentication		HMAC-SHA-1: (A4306) HMAC-SHA2-256: (A4306)
DRBG (Kernel)	DRBG	Random Bit Generation		HMAC DRBG: (A4303) HMAC-SHA2-256: (A4303) SHA2-256: (A4303)
SHA (Kernel)	SHA	Entropy source conditioning component		SHA2-512: (A4303)
Verify image	DigSig-SigVer	Verification of firmware image		ECDSA SigVer (FIPS186-4): (A4302) Curve: P-256 SHA2-256: (A4302)
Entropy Source	ENT-ESV	Entropy source		SHA2-512: (A4303)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

The module includes ECDSA algorithms that have been validated using FIPS 186-4 CAVP tests, which are mathematically identical to FIPS 186-5 CAVP tests. Per IG C.K, all RSA and ECDSA algorithms implemented by the module are claimed compliant with FIPS 186-5.
The module complies with IG C.F. RSA Key Generation, Signature Generation and Signature Verification have been tested and validated using CAVP testing for all implemented modulus lengths (2048, 3072 and 4096 bits). The number of Miller-Rabin tests used for primality testing as part of RSA Key Generation is consistent with Table C.3.

The module implements the following Approved key agreement methods which have been CAVP tested and validated:

- KAS-ECC per SP 800-56A Rev. 3 (FIPS 140-3 IG D.F Scenario 2, path 2).

The module obtains the FIPS 140-3 IG D.F required key agreement assurances in accordance with Section 5.6.2 of SP800-56A Rev. 3. All the key agreement protocols implemented by the module are Diffie-Hellman based.

2.8 RBG and Entropy

The tables below indicate the entropy source used by the module and their associated certificates.

Cert Number	Vendor Name
E103	Juniper Networks

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
EX4100 - Junos OS 22.4 Entropy Source (E103)	Non-Physical	ARM-cortex A72 64-bit, single core	512 bits	448 bits	A4303 (SHA2-512)

Table 11: Entropy Sources

The entropy source is used to seed the module's HMAC DRBG with the minimum required 256-bits of entropy. Each 512-bit block of conditioned output from the entropy source contains 448 bits of entropy. The HMAC DRBG is used for all random data required by the module, including key generation.

There are no initialization procedures required by the users of the module to operate the entropy source in a compliant manner. The module complies with the ESV Public Use document of the validated entropy source (Cert. E103).

2.9 Key Generation

The cryptographic module implements the key generation methods listed above in the Security Functions implementation table.

2.10 Key Establishment

The cryptographic module implements the key establishment methods listed above in the Security Functions implementation table.

2.11 Industry Protocols

The cryptographic module supports the protocols listed below. No part of these protocols, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP. The SSH algorithms allow independent selection of key exchange, authentication, cipher, and integrity. In reference to the supported protocols table below, each column of options for a given protocol is independent and may be used in any viable combination.

Protocol	Key Exchange	Auth	Cipher	Integrity
SSHv2	EC Diffie-Hellman P-256 EC Diffie-Hellman P-384 EC Diffie-Hellman P-521	ECDSA P-256 ECDSA P-384 ECDSA P-521 RSA 2048 RSA 3072 RSA 4096	AES CBC 128/192/256 AES CTR 128/192/256	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-512

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The following table maps each physical interface to one or more logical interface types defined in the FIPS 140-3 standard. The module does not have a Control Output Interface.

Physical Port	Logical Interface(s)	Data That Passes
Ethernet (data)	Data Input Data Output Control Input Status Output	LAN communications
Ethernet (mgmt.)	Data Input Data Output Control Input Status Output	Remote management
Serial	Data Input Data Output Control Input Status Output	Console serial port management
Power	Power	Power
Reset button	Control Input	Reset
USB	Data Input Control Input	Firmware load port
LED	Status Output	Status indicator lighting
SFP28	Data Input Data Output Control Input Status Output	Virtual chassis ports

Table 12: Ports and Interfaces

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module implements two forms of role-based authentication methods, as described in the following table.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Password authentication	User and CO authentication via SSH or consol. Minimum of 10 ASCII character passwords.	SHA (LibMD)	Probability of guessing: 1/(96^10) < 1/1,000,000.	Timed access mechanism allows max of 10 attempts / min. Probability of guessing: 10/(96^10) < 1/100,000.
Signature authentication	User/CO authentication via SSH	SigVer (SSH)	Strength of signature algorithm, minimum 112-bits. Probability of success for random attempt: 1/(2^112) < 1/1,000,000.	A rate of 1 CPU cycle per failed authentication for the ARM-cortex A72 processor (2.2 GHz) allows for the probability of success by brute-force attack: 60 x 2.2 x 10^9 x 1/(2^112) < 1/100,000.

Table 13: Authentication Methods

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	Password authentication Signature authentication
User	Role	Monitor	Password authentication Signature authentication

Table 14: Roles

The module supports two roles: Cryptographic Officer (CO) and User. The module supports concurrent operators but does not support a maintenance role and/or bypass capability. The module enforces the separation of roles using either of the role-based operator authentication methods in Section 4.1.

The Cryptographic Officer role configures and monitors the module via a console or SSH connection. As root or super-user, the Cryptographic Officer has permission to view and edit secrets within the module.

The User role monitors the module via the console or SSH. The user role cannot change the configuration.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure Security	Security relevant configuration	':fips' suffix in CLI prompt	CLI Command	Status	SHA (Kernel) Entropy Source KeyGen (SSH) SHA (LibMD) MAC (LibMD) DRBG (Kernel)	Crypto Officer - HMAC DRBG V value: E - HMAC DRBG Key value: E - HMAC DRBG Entropy Input: E - HMAC DRBG Seed: E - User-PW: W - CO-PW: W - Root-PW: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH PUB: G,R,W - SSH PHK: G,R,W
Configure	Non-security relevant configuration	None	CLI Command	Status	None	Crypto Officer
Show status	Show status	None	None	':fips' suffix in CLI prompt	None	Crypto Officer User
Zeroize	Zeroize all CSPs	None	CLI command	None (completion indicator is implicitly provided by the module rebooting)	None	Crypto Officer - HMAC DRBG V value: Z - HMAC DRBG Key value: Z - HMAC DRBG Entropy Input: Z - HMAC DRBG Seed: Z - SSH DH Shared Secret: Z - SSH PHK: Z - SSH PUB: Z - SSH DH PRV: Z - SSH DH PUB: Z - SSH DH Pub (peer): Z - SSH-SEKs: Z - CO-PW: Z - Root-PW: Z - User-PW: Z - Auth-CO Pub: Z - Auth-User Pub: Z - Root-CA: Z - Package-CA: Z
SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	':fips' suffix in CLI prompt	SSH packets	SSH packets, Status	Enc/Dec (SSH) KAS-SSC (SSH) SigGen (SSH) SigVer (SSH) MAC (SSH) KAS KeyGen (SSH) KDF (SSH) Full KAS (SSH) KTS (SSH) SHA (Kernel) Entropy Source	Crypto Officer - HMAC DRBG V value: E - HMAC DRBG Key value: E - HMAC DRBG Entropy Input: E - HMAC DRBG Seed: E - SSH DH Shared Secret: G,E - SSH DH PRV: G,E - SSH DH PUB: G - SSH-SEKs: G,E - SSH DH Pub (peer): E - CO-PW: E User - HMAC DRBG V value: E - HMAC DRBG Key value: E - HMAC DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Entropy Input: E - HMAC DRBG Seed: E - SSH DH Shared Secret: G,E - SSH DH PRV: G,E - SSH DH PUB: G - SSH-SEKs: G,E - SSH DH Pub (peer): E - User-PW: E
Console access	Console monitoring and control (CLI)	None	CLI Command	Status	None	Crypto Officer - CO-PW: E - Root-PW: E User - User-PW: E
Remote reset	Software initiated reset, performs self-tests on demand.	None	CLI command	Status	None	Crypto Officer - HMAC DRBG V value: Z - HMAC DRBG Key value: Z - HMAC DRBG Entropy Input: Z - HMAC DRBG Seed: Z - SSH DH Shared Secret: Z - SSH DH PRV: Z - SSH DH PUB: Z - SSH-SEKs: Z - SSH DH Pub (peer): Z
Local reset	Hardware reset or power cycle	None	Main power cycle	Status	None	Unauthenticated - HMAC DRBG V value: Z - HMAC DRBG Key value: Z - HMAC DRBG Entropy Input: Z - HMAC DRBG Seed: Z - SSH DH Shared Secret: Z - SSH DH PRV: Z - SSH DH PUB: Z - SSH-SEKs: Z - SSH DH Pub (peer): Z
Traffic	Traffic requiring no cryptographic services	None	Traffic in	Traffic out	None	Unauthenticated
Load Image	Loading of firmware image	':fips' suffix in CLI prompt	CLI Command	Status	Verify image	Crypto Officer - Root-CA: E - Package-CA: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform self-test	On demand execution of all pre-operational and conditional algorithm self-tests	None	Local or remote reset	Status	None	Crypto Officer User Unauthenticated
Show module version	Show system information identifying module	None	CLI command	Status	None	Crypto Officer User

Table 15: Approved Services

4.4 Non-Approved Services

The module does not offer any non-approved services.

N/A for this module.

4.5 External Software/Firmware Loaded

The module includes a firmware load service that is used to install the Junos OS firmware image as part of installation of the module, as described in Section 11.1. The loaded firmware is a complete image replacement and constitutes an entirely new module and version of Junos OS which would require a separate FIPS 140-3 validation.

5 Software/Firmware Security

5.1 Integrity Techniques

The cryptographic module implements a firmware integrity self-test that uses ECDSA P-256 with SHA2-256 to ensure the integrity of all Junos OS firmware components. The self-test is automatically run on power-up.

5.2 Initiate on Demand

The firmware integrity test can be run on demand by the module's operator by power cycling the module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

The module consists of hardware containing a non-modifiable operational environment as per the FIPS 140-3 definitions. It includes a firmware load service to support necessary updates. The loaded firmware is a complete image replacement and constitutes an entirely new module and version of Junos OS which would require a separate FIPS 140-3 validation.

6.2 Configuration Settings and Restrictions

There are no security rules, settings, or restrictions to the configuration of the operational environment beyond the initialization instructions to set the module in Approved mode.

7 Physical Security

The module's physical embodiment meets Level 1 Physical Security requirements. The module is completely enclosed in a rectangular nickel or clear zinc coated, cold rolled steel, plated steel and brushed aluminum enclosure. There are no ventilation holes, gaps, slits, cracks, slots, or crevices that would allow for any sort of observation of any component contained within the cryptographic boundary.

Mechanism	Inspection Frequency	Inspection Guidance
Opaque metal enclosure	n/a	n/a

Table 16: Mechanisms and Actions Required

8 Non-Invasive Security

This section is not applicable, as there are currently no approved non-invasive mitigation techniques specified in ISO/IEC 19790:2012.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The table below lists the areas within the module's cryptographic boundary where SSPs can be stored.

Storage Area Name	Description	Persistence Type
RAM	Random Access Memory	Dynamic
Flash	Internal flash memory storage drive	Static

Table 17: Storage Areas

9.2 SSP Input-Output Methods

The table below lists the method used by the module for the input and output of SSPs.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Entry via SSH	Remote CO	RAM	Encrypted	Automated	Electronic	KTS (SSH)
Entry via console	Local CO	RAM	Plaintext	Manual	Electronic	
Output via SSH	RAM	Remote CO	Encrypted	Automated	Electronic	KTS (SSH)
Output via console	RAM	Local CO	Plaintext	Manual	Electronic	

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Entry as part of KAS	Remote peer	RAM	Plaintext	Automated	Electronic	Full KAS (SSH)
Output as part of KAS	RAM	Remote peer	Plaintext	Automated	Electronic	Full KAS (SSH)
Pre-loaded	Manufacturer	Flash	Plaintext	Manual	Direct	

Table 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The table below describes the SSP zeroization methods employed by the module.

Zeroization Method	Description	Rationale	Operator Initiation
Zeroize CLI command	This command erases all data, including all configuration information, returning the module to its factory default state The system is then rebooted.	This command erases all keys and CSPS from storage. The forced power cycle also zeroizes SSPs in volatile memory.	Yes, CO via invocation of zeroize CLI command.
Reset	Zeroization of SSPs in RAM via invocation of local or remote reset service.	RAM is volatile and all data is lost when power is taken off. Zeroization is practically instantaneous.	Yes, both User and CO, via invocation of Local Reset or Remote Reset services.
Explicit zeroize function	Zeroization of SSPs in memory when no longer needed.	Use of explicit zeroization function destroys SSP information immediately by overwriting memory area with zeroes.	No. The operator cannot directly initiate this method.

Table 19: SSP Zeroization Methods

The Zeroize CLI command method is detailed in section 11.2.3.

The completion of zeroization is indicated implicitly. If the zeroization is initiated using a zeroization command or explicit delete command, completion of the command indicates that zeroization has successfully completed. If the zeroization is initiated by power cycling the module, then successful reboot of the module indicates that zeroization has completed successfully. In the case of zeroization initiated by session termination, SSPs are zeroized when the session terminates, and session termination is indicated in the log.

9.4 SSPs

All SSPs used by the module are described in this section.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC DRBG V value	A critical value of the internal state of DRBG	256 - 256	DRBG internal state - CSP	DRBG (Kernel)		DRBG (Kernel)
HMAC DRBG Key value	A critical value of the internal state of DRBG	256 - 256	DRB internal state - CSP	DRBG (Kernel)		DRBG (Kernel)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC DRBG Entropy Input	A critical value of the internal state of DRBG provided by entropy source	256 - 256	Entropy source output - CSP	Entropy Source		DRBG (Kernel)
HMAC DRBG Seed	Seed material used to seed or reseed the HMAC DRBG	256 - 256	DRBG internal state - CSP	DRBG (Kernel)		DRBG (Kernel)
SSH DH Shared Secret	Shared DH value computed from the ephemeral DH key-pairs as part of SSH and used to derive session keys.	256, 384, 521 - 128, 192, 256	DH shared value - CSP		KAS-SSC (SSH)	KDF (SSH)
SSH PHK	SSH Private host key. 1st time SSH is configured, the keys are generated.	2048, 256, 4096, 384, 521 - 112, 128, 152, 192, 256	Asymmetric private key - CSP	KeyGen (SSH)		SigGen (SSH)
SSH PUB	SSH Public Host Key	2048, 256, 4096, 384, 521 - 112, 128, 152, 192, 256	Asymmetric public key - PSP	KeyGen (SSH)		SigVer (SSH)
SSH DH PRV	SSH KAS private key	256, 384, 521 - 128, 192, 256	Asymmetric private key - CSP	KAS KeyGen (SSH)		KAS-SSC (SSH) Full KAS (SSH)
SSH DH PUB	SSH KAS public key	256, 384, 521 - 128, 192, 256	Asymmetric public key - PSP	KAS KeyGen (SSH)		
SSH DH Pub (peer)	SSH KAS public key from peer	256, 384, 521 - 128, 192, 256	Asymmetric public key - PSP			KAS-SSC (SSH) Full KAS (SSH)
SSH-SEKs	SSH Session Encryption Keys	128, 192, 256 - 128, 192, 256	Symmetric key - CSP		KDF (SSH) Full KAS (SSH)	Enc/Dec (SSH) MAC (SSH)
CO-PW	Password used to authenticate the CO.	Min 10 characters - n/a	Authentication password - CSP		KTS (SSH)	SHA (LibMD)
Root-PW	Password used by CO to authenticate as 'root'.	Min 10 characters - n/a	Authentication password - CSP		KTS (SSH)	SHA (LibMD)
User-PW	Password used to authenticate User	Min 10 characters - n/a	Authentication password - CSP		KTS (SSH)	SHA (LibMD)
Auth-CO Pub	SSH CO Authentication Public Key	2048, 4096, 256, 384, 521 - 112, 128, 152, 192, 256	Asymmetric public key - PSP		KTS (SSH)	SigVer (SSH)
Auth-User Pub	SSH User Authentication Public Key	2048, 4096, 256, 384, 521 - 112, 128, 152, 192, 256	Asymmetric public key - PSP		KTS (SSH)	SigVer (SSH)
Root-CA	X.509 Certificate used to verify the validity of the Juniper Package CA	256, 384 - 128, 196	Asymmetric public key - PSP			Verify image
Package-CA	X.509 Certificate used to verify the validity the	256 - 128	Asymmetric public key - PSP			Verify image

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	Juniper Image at software load and also at runtime for integrity.					

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
HMAC DRBG V value		RAM:Plaintext	Until updated by HMAC_DRBG_Update()	Zeroize CLI command Reset	
HMAC DRBG Key value		RAM:Plaintext	Until updated by HMAC_DRBG_Update()	Zeroize CLI command Reset	
HMAC DRBG Entropy Input		RAM:Plaintext	Until HMAC_Instantiate_Update() or HMAC_DRBG_Reseed() complete	Zeroize CLI command Reset	
HMAC DRBG Seed		RAM:Plaintext	Until HMAC_Instantiate_Update() or HMAC_DRBG_Reseed() complete	Zeroize CLI command Reset	
SSH DH Shared Secret		RAM:Plaintext	Until SSH session termination	Zeroize CLI command Reset Explicit zeroize function	
SSH PHK	Entry via SSH Entry via console Output via SSH Output via console	RAM:Plaintext Flash:Plaintext	Until SSH session termination (RAM)	Zeroize CLI command	SSH PUB:Paired With
SSH PUB	Entry via SSH Entry via console Output via SSH Output via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	SSH PHK:Paired With
SSH DH PRV		RAM:Plaintext	Until SSH session termination	Reset Explicit zeroize function	SSH DH PUB:Paired With
SSH DH PUB	Output as part of KAS	RAM:Plaintext	Until SSH session termination	Reset Explicit zeroize function	SSH DH PRV:Paired With
SSH DH Pub (peer)	Entry as part of KAS	RAM:Plaintext	Until SSH session termination	Reset Explicit zeroize function	
SSH-SEKs		RAM:Plaintext	Until SSH session termination	Reset Explicit zeroize function	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
CO-PW	Entry via SSH Entry via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
Root-PW	Entry via SSH Entry via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
User-PW	Entry via SSH Entry via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
Auth-CO Pub	Entry via SSH Entry via console Output via SSH Output via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
Auth-User Pub	Entry via SSH Entry via console Output via SSH Output via console	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
Root-CA	Pre-loaded	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	
Package-CA	Pre-loaded	RAM:Plaintext Flash:Plaintext		Zeroize CLI command	

Table 21: SSP Table 2

9.5 Transitions

The following transitions apply to algorithms used by this module:

SHA-1: The SHA-1 hash algorithm will be non-Approved for cryptographic protection purposes after December 31, 2030.

10 Self-Tests

On power up or reset, the module performs the pre-operational self-tests and the indicated conditional cryptographic algorithm self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. The CASTs for algorithms utilized in the pre-operational Firmware integrity check are performed prior to the Firmware integrity check.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware integrity check	ECDSA P-256 with SHA2-256	KAT	SW/FW Integrity	PASS/FAIL console output	ECDSA verify
Critical functions test	SHA2-256	KAT	Critical Function	PASS/FAIL console output	Checks that any file that is executed is registered in a manifest of executable files that comes with the firmware. Test verifies the integrity of the operational environment is being enforced by having the kernel attempt to run a specific executable file that does not contain a hash in the manifest file, verifying it cannot be executed.

Table 22: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Entropy Source (start-up)	n/a	APT, RCT	CAST	PASS/FAIL console output	Start-up	On-power up
Entropy Source (continuous)	n/a	APT, RCT	CAST	Console output / output of entropy source	Continuous	Data output from noise source
AES-CBC (A4301) Encrypt	Key size: 128, 192, 256	KAT	CAST	PASS/FAIL console output	Encrypt	On power-up
AES-CBC (A4301) Decrypt	Key size: 128, 192, 256	KAT	CAST	PASS/FAIL console output	Decrypt	On power-up
HMAC-SHA-1 (A4301)	Key size: 160	KAT	CAST	PASS/FAIL console output	MAC	On power-up
HMAC-SHA2-256 (A4301)	Key size: 256	KAT	CAST	PASS/FAIL console output	MAC	On power-up
HMAC-SHA2-384 (A4301)	Key size: 384	KAT	CAST	PASS/FAIL console output	MAC	On power-up
HMAC-SHA2-512 (A4301)	Key size: 512	KAT	CAST	PASS/FAIL console output	MAC	On power-up
RSA SigGen (FIPS186-5) (A4301)	RSA 2048 w/ SHA2-256, RSA 4096 w/ SHA2-256	KAT	CAST	PASS/FAIL console output	Sign	On power-up
RSA SigVer (FIPS186-5) (A4301)	RSA 2048 w/ SHA2-256, RSA 4096 w/ SHA2-256	KAT	CAST	PASS/FAIL console output	Verify	On power-up
ECDSA SigGen (FIPS186-4) (A4301)	P-256, P-384, P-521	KAT	CAST	PASS/FAIL console output	Sign	On power-up
ECDSA SigVer (FIPS186-4) (A4301)	P-256, P-384, P-521	KAT	CAST	PASS/FAIL console output	Verify	On power-up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 (A4301)	P-256, P-384, P-521	KAT	CAST	PASS/FAIL console output	ECDH Computation	On power-up
KDF SSH (A4301)	SHA-1, SHA2-256, SHA2-384	KAT	CAST	PASS/FAIL console output	Key derivation Computation	On power-up
RSA KeyGen (FIPS186-5) (A4301)	n/a	PCT	PCT	Returned key/transition soft error state	Generation and Verification of signature	On key generation
ECDSA KeyGen (FIPS186-4) (A4301)	n/a	PCT	PCT	Returned key/transition soft error state	Generation and Verification of signature	On key generation
ECDSA SigVer (FIPS186-4) (A4302)	P-256	KAT	CAST	PASS/FAIL console output	Verify	On power-up
FW Load	ECDSA P-256 with SHA2-256	KAT	SW/FW Load	PASS/FAIL console output	Verification of ECDSA signature on FW	On FW load
HMAC DRBG (A4303)	256, SHA2-256	KAT	CAST	PASS/FAIL console output	Health-tests initialise, re-seed, and generate	On power-up
HMAC-SHA-1 (A4303)	Key size: 160	KAT	CAST	PASS/FAIL console output	MAC	On power-up
HMAC-SHA2-256 (A4303)	Key size: 256	KAT	CAST	PASS/FAIL console output	MAC	On power-up
SHA2-384 (A4303)	n/a	KAT	CAST	PASS/FAIL console output	Hash	On power-up
SHA2-512 (A4303)	n/a	KAT	CAST	PASS/FAIL console output	Hash	On power-up
HMAC-SHA2-256 (A4306)	Key size: 256	KAT	CAST	PASS/FAIL console output	MAC	On power-up
HMAC-SHA-1 (A4306)	Key size: 256	KAT	CAST	PASS/FAIL console output	MAC	On power-up
SHA2-512 (A4306)	n/a	KAT	CAST	PASS/FAIL console output	Hash	On power-up

Table 23: Conditional Self-Tests

10.3 Periodic Self-Test Information

The module does not implement periodic self-testing.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware integrity check	KAT	SW/FW Integrity	On demand	Manually
Critical functions test	KAT	Critical Function	On demand	Manually

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Entropy Source (start-up)	APT, RCT	CAST	On demand	Manually
Entropy Source (continuous)	APT, RCT	CAST	Continuous	Automatically
AES-CBC (A4301) Encrypt	KAT	CAST	On Demand	Manually
AES-CBC (A4301) Decrypt	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4301)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4301)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4301)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A4301)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4301)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4301)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4301)	KAT	CAST	On Demand	Manually
KDF SSH (A4301)	KAT	CAST	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4301)	PCT	PCT	On trigger condition	Automatic
ECDSA KeyGen (FIPS186-4) (A4301)	PCT	PCT	On trigger condition	Automatic
ECDSA SigVer (FIPS186-4) (A4302)	KAT	CAST	On Demand	Manually
FW Load	KAT	SW/FW Load	On FW load request	Automatic
HMAC DRBG (A4303)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4303)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4303)	KAT	CAST	On Demand	Manually
SHA2-384 (A4303)	KAT	CAST	On Demand	Manually
SHA2-512 (A4303)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4306)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4306)	KAT	CAST	On Demand	Manually
SHA2-512 (A4306)	KAT	CAST	On Demand	Manually

Table 25: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Critical Failure State	The cryptographic module ceases to perform cryptographic operations, inhibits all data output, and provides status of the error via syslog messages and console status output	On any power-up self-test or PCT failure	Power cycle	Console status indicator
Soft Error State	A non-critical self-test failure occurs, causing a failure of the triggering operation	Firmware load test or continuous entropy health test failure	The module processes the error, and resumes normal operation	Console displays error

Table 26: Error States

The module enters critical failure state upon failure of a self-test, causing the kernel to ‘panic‘ and all execution to halt. The only way to exit from this state is to reboot the module, which causes the self-tests to be repeated and pass successfully before the corresponding algorithms are usable.

10.5 Operator Initiation of Self-Tests

Self–tests that are performed at power-up are available on demand by power cycling the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module must be correctly installed and configured to enter a FIPS compliant state and operate in the Approved mode. The required procedures are as follows:

1. Install the Junos OS firmware image - the procedure is detailed in section 11.2.1
2. Configure device for the Approved mode - the procedure is section 11.2.2.

To continue using the module in a FIPS compliant way, the Module Operation Rules in section 11.4.2 must be followed.

11.2 Administrator Guidance

11.2.1 Installing the Junos OS firmware image

1. Download the validated firmware image from https://www.juniper.net/support/downloads/junos.html. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by Juniper Networks representatives. Select the validated firmware image. Download the firmware image to a local host or to an internal software distribution site.

   The cryptographic module devices use the following firmware image:

   junos-install-ex-arm-64-22.4R2.8.tgz
2. Connect to the console port on the device from your management device, and log in to the Junos OS CLI.

3. Install the new package on the device (package may be a local file copied to the device, or a file on a remote server):
   ```
   user@host> request system software add <package>
   ```
4. Reboot the device to load the installation:
   ```
   user@host> request system reboot
   ```
5. After the reboot has completed, log in and use the show version command to verify that the new version of the software is successfully installed.
   ```
   user@host> request vmhost reboot
   ```

11.2.2 Configure the device for the Approved mode

To configure the device for the Approved mode:

1. Zeroize the device to delete all CSPs before entering the Approved mode.
   ```
   root@host# request system zeroize
   ```
2. After the device comes up, login using username “root” and password blank.
3. Configure root authentication with password at least 10 characters or more.
   ```
   root@host# set system root-authentication plain-text-password
   ```
4. Load configuration onto device and commit new configuration.

   NOTE: SSH key-exchange configuration must not include ‘dh-group14-sha1’. It is not approved for this module.
5. Configure crypto-officer and login with crypto-officer credentials.
6. Set the fips level to 1.
   ```
   crypto-officer@host# set system fips level 1
   ```
7. Commit and reboot the device.
   ```
   crypto-officer@host# commit
   crypto-officer@host# run request system reboot
   ```

11.2.3 Zeroizing the System

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

After zeroizing the system, the module is no longer in a FIPS compliant state. (Installation and configuration as per section 11.1 is required to enter the FIPS compliant state and enable the Approved mode of operation).

NOTE: The Crypto-Officer must retain control of the module while zeroization is in progress.

To zeroize the device:

1. Login to the device as Crypto Officer and from CLI, enter
   ```
   crypto-officer@host# request system zeroize
   ```

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

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

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

11.3 Non-Administrator Guidance

No specific non-administrator guidance is required to operate the module.

11.4 Design and Rules

11.4.1 Module Design Rules

The module design implements the following security rules:

1. The module clears previous authentications on power cycle.
2. Power up self-tests do not require any operator action.
3. Data output is inhibited during key generation, self-tests, zeroization, and error states.
4. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
5. There are no restrictions on which SSPs are zeroized by the zeroization service.
6. The module does not support a maintenance interface or role.
7. The module does not output intermediate key values.
8. The module requires two independent internal actions to be performed prior to outputting plaintext CSPs.

11.4.2 Module Operation Rules

The following are requirements for compliant usage of the module:

1. The cryptographic officer must retain control of the module while zeroization is in process.
2. The cryptographic officer shall verify that the firmware image to be loaded on the module is a FIPS validated image.
3. Before pushing the factory reset button on the device, the cryptographic officer shall perform the zeroize command as described in section 11.2.3.
4. The password minimum-length must be configured to be at least 10.
5. Virtual Chassis features must not be configured.
6. SSH key-exchange must not be configured to include 'dh-group14-sha1'.

11.5 Maintenance Requirements

No special maintenance requirements are required.

11.6 End of Life

When disposing of the cryptographic module, the cryptographic officer shall perform the zeroize command as described in Section 11.2.3.

12 Mitigation of Other Attacks

The module does not implement mechanisms to mitigate other attacks beyond what is described in this security policy.