Juniper Networks, Inc.

Juniper Networks PTX10008 and PTX10016 Packet Transport Routers

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

Introduction
Federal Information Processing Standards Publication 140-3 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140-3 program. The NVLAP accredits independent testing labs to perform FIPS 140-3 testing; the CMVP validates modules meeting FIPS 140-3 validation. Validated is the term given to a module that is documented and tested against the FIPS 140-3 criteria.

More information is available on the CMVP website at:
https://csrc.nist.gov/projects/cryptographic-module-validation-program.

About this Document
This non-proprietary Cryptographic Module Security Policy for the Juniper Networks PTX10008 and PTX10016 Packet Transport Routers with Routing Engine JNP10K-RE0 and MACsec Line Card LC1105-M provides an overview of the product and a high-level description of how it meets the overall Level 1, security requirements of FIPS 140-3.

The Juniper Networks PTX series router models: the PTX10008 and PTX10016 may also be referred to as the “module” in this document.

Disclaimer
The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Juniper Networks shall have no liability for any error or damages of any kind resulting from the use of this document.

Notices

This document describes the cryptographic module security policy for the Juniper Networks PTX series router models: the PTX10008 and PTX10016 cryptographic module (also referred to as the “module” hereafter) with firmware version Junos OS 22.4R2.8. The module has a multi-chip standalone embodiment. It contains specification of the security rules, under which the cryptographic module operates, including the security rules derived from the requirements of the FIPS 140-3 standard.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	3
5	Software/Firmware security	1

Section	Title	Security Level
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

1.3 Additional Information

The module claims an overall Security Level of 1 with all individual sections at a Security Level 1 with the exceptions of Roles, Services and Authentication (claimed at Security Level 3). The module does not implement any non-invasive security mitigations or mitigations of other attacks and thus the requirements per these sections are inapplicable.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The cryptographic module provides for an encrypted connection, using SSH, between the management station and itself, i.e., the PTX series routers. The cryptographic module also provides for an encrypted connection, using MACsec, between itself and a peer.

Module Type: Hardware

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The cryptographic module's operational environment is a limited operational environment. The cryptographic boundary of the hardware module is the entirety of the module/chassis. This includes the Routing Engine (RE). No components have been excluded from the cryptographic boundary of the module.

Tested Operational Environment's Physical Perimeter (TOEPP):

The Tested Operational Environment's Physical Perimeter (TOEPP) is the entirety of the module chassis.

The images below depict the physical boundary of the modules, including the Routing Engine, the PTX10KLC1105-M MACsec Line Card and SIB. The non-crypto-relevant line cards included in the figure are not inserted in the module/excluded from the boundary per the scope of this validation.

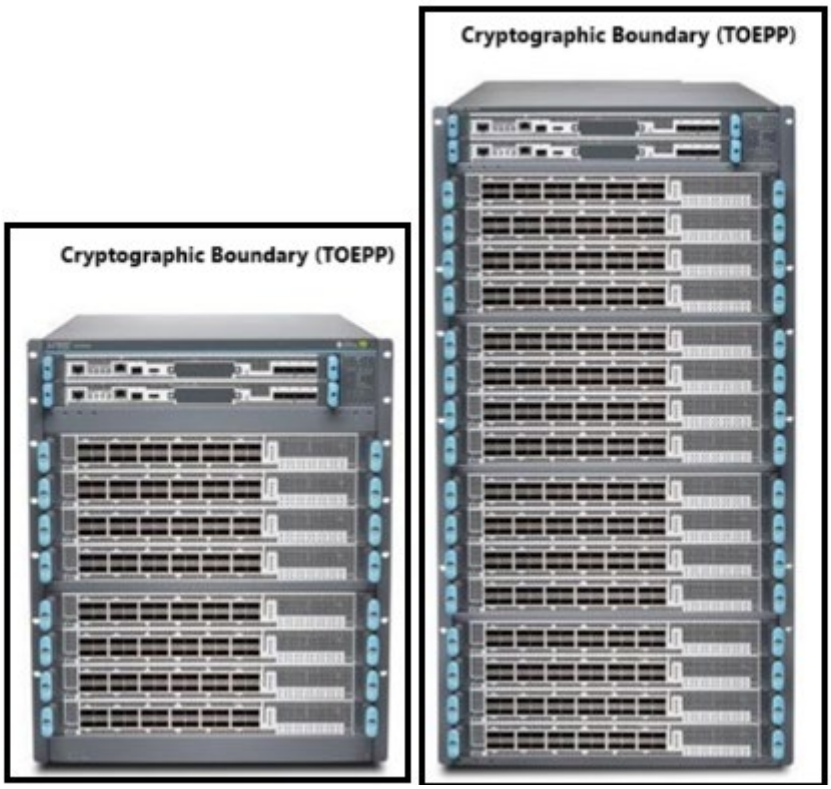

Figure 1: Physical Cryptographic Boundary [Left to Right: PTX10008 and PTX10016]

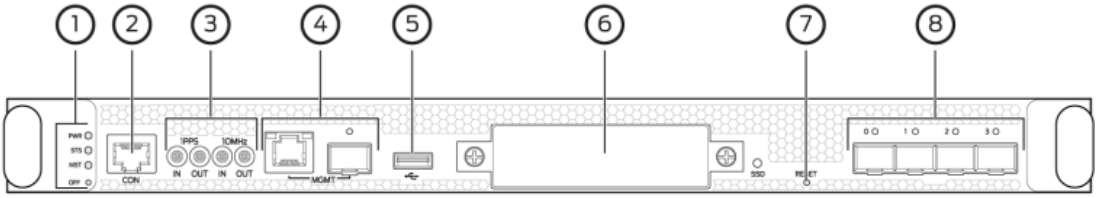

Figure 2: Routing Engine [JNP10K-RE0]

1- RCB status LEDs
2- Console port (CON)
3- PTP-capable connections: SMB In, SMB Out, 10 MHz In, 10 MHz Out
4- Management port (MGMT)
5- USB 2.0 port
6- Secondary 50-GB SATA SSD slot
7- Reset (RESET) button
8- Four SFP+ ports (reserved for future use)

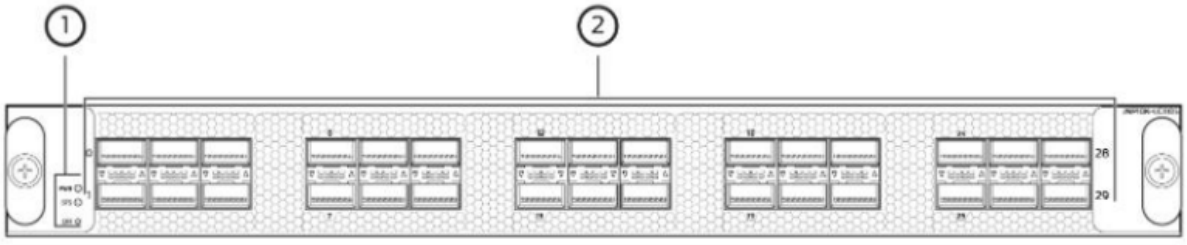

Figure 3: PTX10K-LC1105-M MACSec Line Card

1- Power LED (PWR), status LED (STS), and offline (OFF) button
2- Network ports.

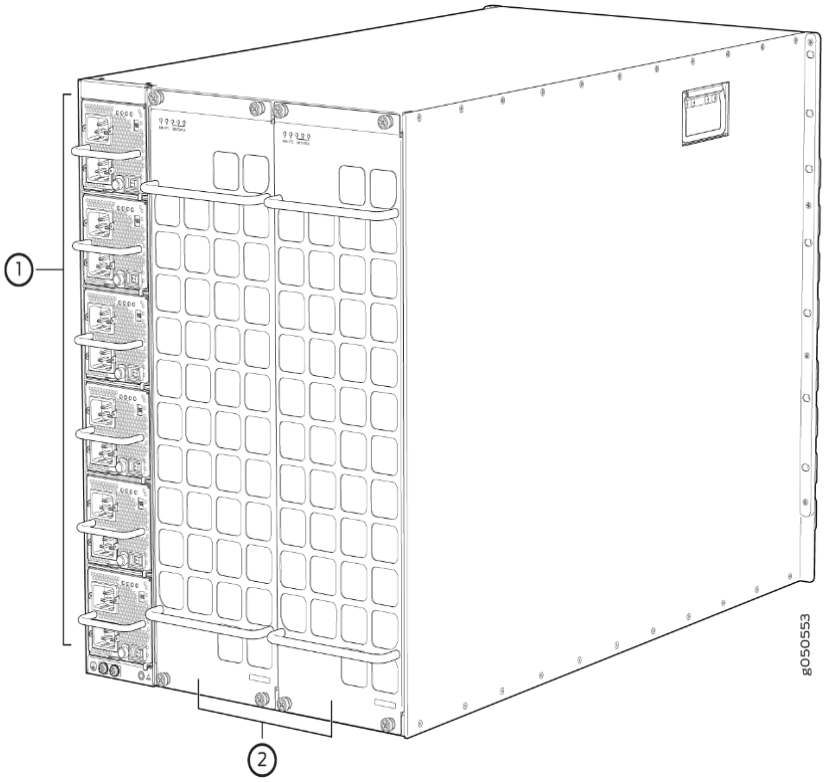

Figure 4: PTX10008 Chassis Rear

1- AC or DC power supplies numbered 0–5 (top to bottom)
2- Fan trays with redundant fans

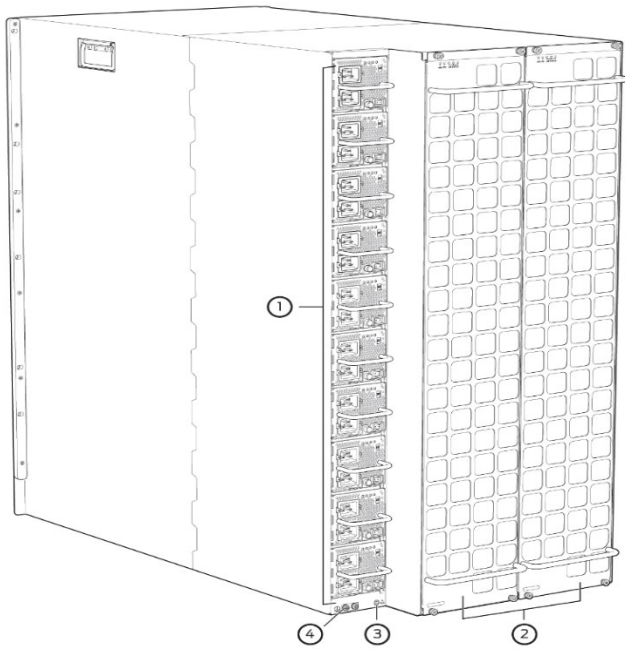

Figure 5: PTX10016 Chassis Rear

1- Power supplies

2- Fan trays

3- ESD point

4- Protective earthing terminal

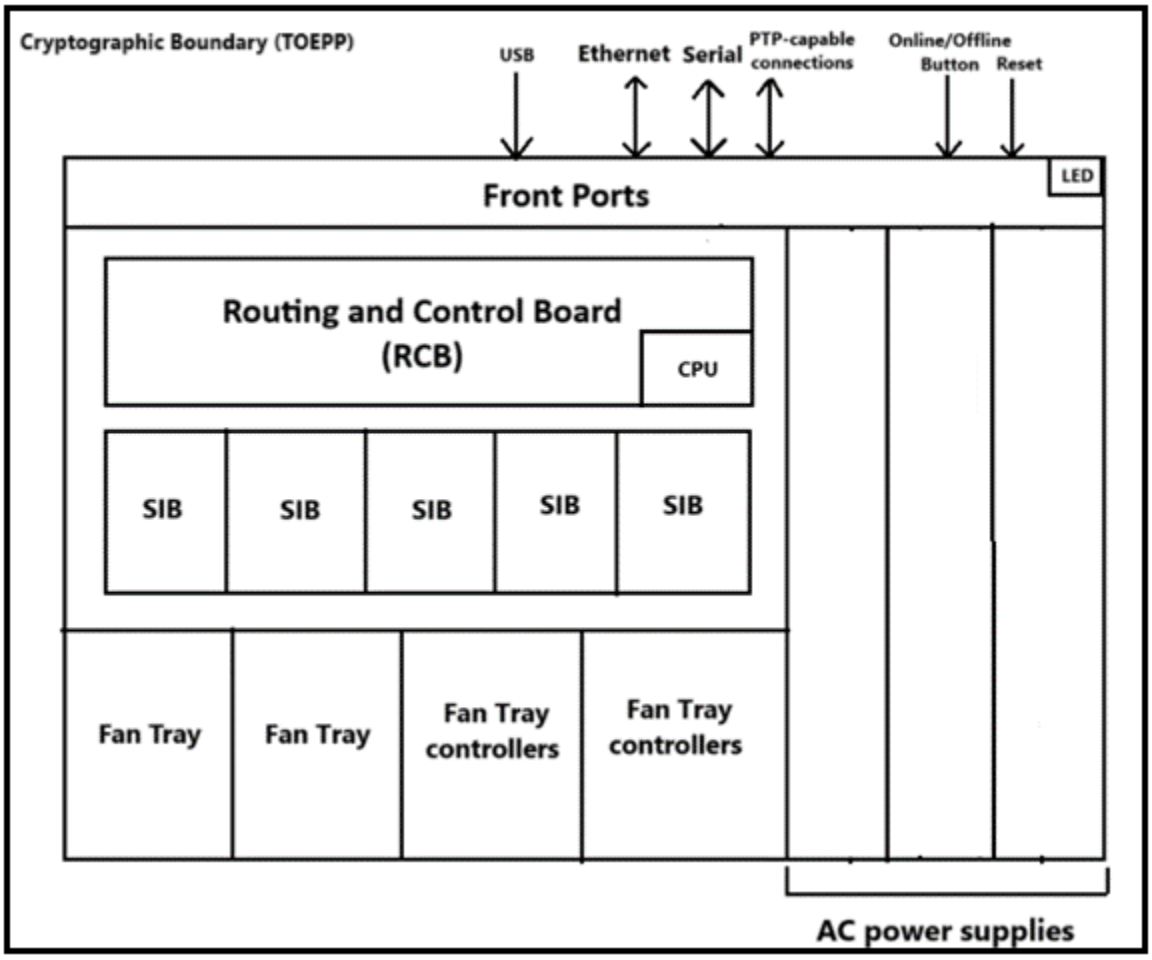

Figure 6: PTX10008 Block diagram

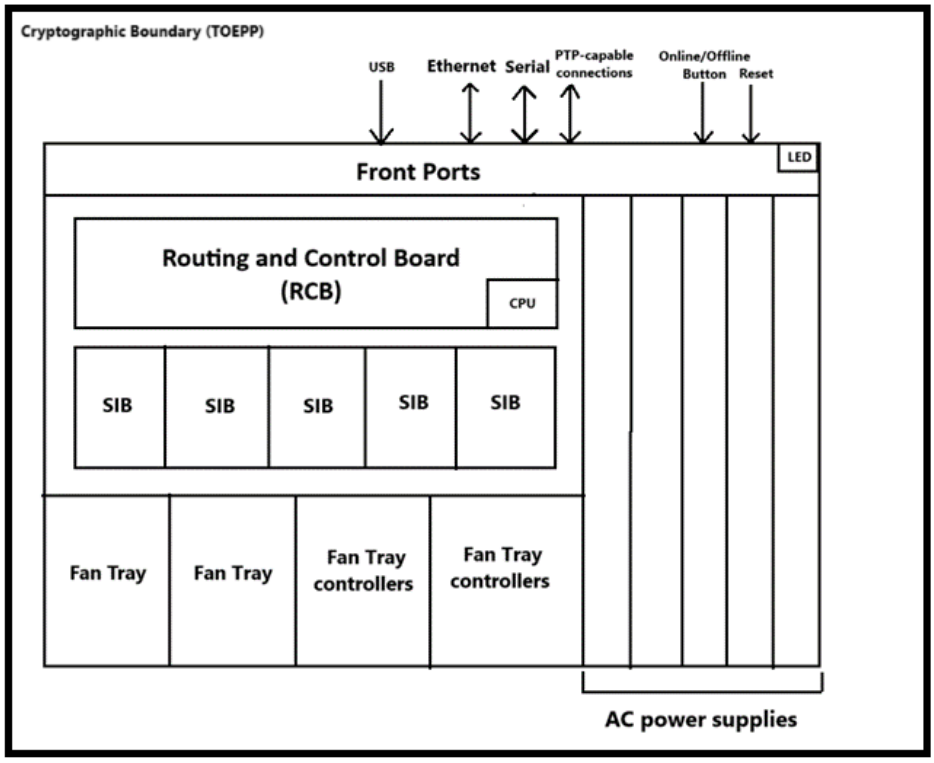

Figure 7: PTX10016 Block diagram

The PTX10K-LC1105-M has thirty 28-Gbps QSFP+ (QSFP28) ports that are Media Access Control Security (MACsec) capable, each of which can be configured via the CLI to support speeds of 100 Gbps or 40 Gbps.

Each of the 30 QSFP28 ports can operate as:
• 100-Gigabit Ethernet ports when using QSFP28 optical transceivers.
• 40-Gigabit Ethernet ports when using QSFP+ optical transceivers.

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
PTX10008	PTX10008	Junos OS 22.4R2.8	Intel Xeon E3-1125v2	JNP10K-PWR-AC; JNP10K-PWR-DC; JNP10K-PWRAC2; JNP10K-PWR-DC2

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
PTX10016	PTX10016	Junos OS 22.4R2.8	Intel Xeon E3-1125v2	JNP10K-PWR-AC; JNP10K-PWR-DC; JNP10K-PWRAC2; JNP10K-PWR-DC2

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

No components have been excluded from the cryptographic boundary of the module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	• The operator can verify that the cryptographic module is in the Approved mode by observing the console prompt and running the “show version” command; • When operating in the Approved mode, the prompt will read “<operator>:fips#” (e.g. root:fips#); • The “show version” command will allow the Crypto Officer to verify that the validated firmware version is running on the module; • The Crypto Officer can also use the “show system fips chassis level” command (returns “level 1”) to determine if the module is operating in the Approved mode; • The Approved mode is entered when the module is configured for it and successfully passes all self-tests (both pre-operational and conditional cryptographic algorithm self-tests (CASTs))	Approved	global indicator (string 'fips' included in the command prompt)
Non-Approved mode	• The cryptographic module supports a non-Approved mode of operation; • When operated in the non-Approved mode of operation, the module supports non-Approved algorithms as well as the algorithms supported in the Approved mode of operation	Non-Approved	global indicator (implicit indicator based on exclusion of string 'fips' from the command prompt)

Table 3: Modes List and Description

The hardware versions contained in Table 2, with Junos OS 22.4R2.8 installed, contain one Approved mode of operation and a non-Approved mode of operation. The Junos OS 22.4R2.8

firmware image must be installed on the module. The module is configured during initialization by the Crypto Officer to operate in the Approved mode or the non-Approved mode.

When operated in the non-Approved mode of operation, the module supports non-Approved algorithms as well as the algorithms supported in the Approved mode of operation. The module is in a non-compliant state by default and the Crypto Officer can place the module into the non-Approved mode of operation by following the instructions in Section 11 Life-Cyle Assurance in this document.

Mode Change Instructions and Status:

The module must always be zeroised when switching between the Approved mode of operation and the non-Approved mode of operation and vice versa.

Degraded Mode Description:

The module does not support a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4301	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A4304	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A4304	Direction - Generation, Verification Key Length - 128, 256	SP 800-38B
AES-CTR	A4301	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4301	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4304	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	AES 4369	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 256	SP 800-38D
AES-KW	A4304	Direction - Decrypt, Encrypt Key Length - 128	SP 800-38F
AES-XPN	AES 4369	Direction - Decrypt, Encrypt Key Length - 128, 256 IV Generation - External	SP 800-38D
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

Algorithm	CAVP Cert	Properties	Reference
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 DRBG	A4301	Prediction Resistance - Yes Mode - SHA2-256	SP 800-90A Rev. 1
HMAC DRBG	A4303	Prediction Resistance - Yes Mode - SHA2-256	SP 800-90A Rev. 1
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-SHA2-256	A4303	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
KAS-FFC-SSC Sp800-56Ar3	A4301	Domain Parameter Generation Methods - FC, MODP-2048 Scheme - dhEphem - KAS Role - initiator	SP 800-56A Rev. 3
KDF SP800-108	A4304	KDF Mode - Counter Supported Lengths - Supported Lengths: 128, 256	SP 800-108 Rev. 1
KDF SSH (CVL)	A4301	Cipher - AES-128, AES-192, AES-256, TDES Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-4)	A4301	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A4301	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A4301	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
Safe Primes Key Generation	A4301	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
Safe Primes Key Verification	A4301	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
SHA-1	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A4301	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4303	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
SHA2-512	A4303	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4306	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

The following protocols are supported by the module in the Approved mode:

SSHv2 (EC Diffie-Hellman P-256, P-384, P-521; Diffie-Hellman MODP2048; RSA 2048, 3072, 4096 bits; ECDSA P-256, P-384, P-521; AES CBC 128, 192, 256 bits; AES CTR 128, 192, 256 bits, HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512)

MACsec (MACsec Key Agreement (MKA); AES GCM, XPN 128 and 256 bits)

The SSH protocol allows independent selection of key exchange, authentication, cipher and integrity algorithms. Please note that there are algorithms, modes, and key/moduli sizes that have been CAVP-tested but are not used by any approved service of the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the table above are used by an approved service of the module.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG - Section 4 and 5.1	Key Type:Asymmetric	N/A	NIST SP800-133r2 Section 4: Asymmetric seed generation using an unmodified output from an Approved DRBG; Section 5.1: Key Pairs for Digital Signature Schemes
CKG - Section 4 and 5.2	Key Type:Asymmetric	N/A	NIST SP800-133r2 Section 4: Asymmetric seed generation using an unmodified output from an Approved DRBG; Section 5.2: Key Pairs for Key Establishment
CKG - Section 6.2.1	Key Type:Symmetric	N/A	NIST SP800-133r2 Section 6.2.1: Derivation of symmetric keys

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

Name	Properties	Implementation	Reference
N/A		N/A	N/A

Table 6: Non-Approved, Allowed Algorithms

The module does not support any non-Approved algorithms in the Approved mode, i.e., it does not support Non-Approved Algorithms Allowed in the Approved Mode of Operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
SHA2-256 (Junos 22.4R2 - LibMD Implementation)	no security claimed	Used to store operator passwords in hashed form, per IG 2.4.A: Use of a non-approved cryptographic algorithm to “obfuscate” a CSP
SHA-1 (Junos 22.4R2 - Kernel Implementation)	no security claimed	Used for an extraneous check in the Kernel, per IG 2.4.A: Use of an approved, non-approved or proprietary algorithm for a purpose that is not security relevant

Table 7: Non-Approved, Allowed Algorithms with No Security Claimed

The module does not support any non-Approved algorithms in the Approved mode, i.e., it does not support Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
RSA with key size less than 2048	SSH
ECDSA with ed25519 curve	SSH
EC Diffie-Hellman with ed25519 curve	SSH
ARCFOUR	SSH
Blowfish	SSH
CAST	SSH
DSA (SignGen, SigVer, non-compliant)	SSH
HMAC-MD5	SSH
HMAC-RIPEMD160	SSH
UMAC	SSH

Table 8: Non-Approved, Not Allowed Algorithms

In addition to the above non-Approved Algorithms Not Allowed in the Approved Mode of Operation, all Approved algorithms supported in the Approved mode of operation are also supported in the non-Approved mode.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
KAS1	KAS-135KDF KAS-SSC	Key Agreement for SSHv2	SP 800-56Arev3 KAS-ECC per IG D.F Scenario 2 path (2):size: P-256, P-384, P-521 curves; encryption strength:128, 192, 256 bits; strength caveat: SSP establishment methodology provides between 128 and 256 bits of encryption strength	KAS-ECC-SSC Sp800-56Ar3 KDF SSH
KAS2	AsymKeyPair-KeyGen AsymKeyPair-KeyVer KAS-135KDF KAS-SSC	Key Agreement for SSHv2	SP800-56Arev3 KAS-FFC per IG D.F Scenario 2 path (2):size: MODP 2048; encryption strength: SSP establishment methodology provides 112 bits of encryption strength	KAS-FFC-SSC Sp800-56Ar3 KDF SSH Safe Primes Key Generation Safe Primes Key Verification
KTS1	KTS-Wrap	Key Transport for SSHv2	SP800-38A AES CBC, CTR and HMAC 198 per IG D.G:size: 128, 192, and 256-bit keys; SSP establishment methodology provides between 128 and 256 bits	AES-CBC AES-CTR AES-ECB HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-512 SHA-1 SHA2-256 SHA2-512

Name	Type	Description	Properties	Algorithms
			of encryption strength	
ECDSA SigVer	DigSig-SigVer	ECDSA Signature Verification used for firmware integrity	FIPS 186-4 :size: P-256, encryption strength: 128 bits	ECDSA SigVer (FIPS186-4)
ECDSA SigVer2	DigSig-SigVer	ECDSA Signature Verification used for identity-based public key authentication	FIPS 186-4:size: P-256, P-384, P-521 curves, 128, 192 and 256 bits	ECDSA SigVer (FIPS186-4)
DRBG	DRBG	Kernel DRBG providing random bits to the DRBG2 for SSP generation in the user/application space		HMAC DRBG HMAC-SHA2-256 SHA2-256
DRBG2	DRBG	SSP generation in user/application space		HMAC DRBG HMAC-SHA2-256 SHA2-256
Entropy Souce	ENT-Cond	Non-Physical Entropy Source		SHA2-512
ECDSA KeyGen	AsymKeyPair-KeyGen	Generation of SSH host keys		ECDSA KeyGen (FIPS186-4)
ECDSA KeyGen2	AsymKeyPair-KeyGen	SSP Agreement in the context of SSH		ECDSA KeyGen (FIPS186-4)
ECDSA KeyVer	AsymKeyPair-KeyVer	Verification of keys generated		ECDSA KeyVer (FIPS186-4)
ECDSA SigGen	DigSig-SigGen	Signature Generation using ECDSA in the context of SSH		ECDSA SigGen (FIPS186-4)
RSA KeyGen	AsymKeyPair-KeyGen	Generation of SSH host keys		RSA KeyGen (FIPS186-4)
RSA SigGen	DigSig-SigGen	Signature Generation using RSA in the context of SSH		RSA SigGen (FIPS186-4)
RSA SigVer	DigSig-SigVer	Signature Verification using RSA for public key authentication		RSA SigVer (FIPS186-4)

Name	Type	Description	Properties	Algorithms
Password Hash	SHA	Used to store passwords in hashed form		SHA2-512
CKG	CKG	Cryptographic Key Generation (CKG)		CKG - Section 6.2.1 Key Type: Symmetric
MACsec Encryption/Decryption	BC-Auth	Encryption/Decryption of MACsec packets		AES-GCM AES-XPN
KTS2	KTS-Wrap	Key Transport for MACsec	SP800-38D AES KW per IG D.G :size:128-,192-,256-bit keys; encryption strength: SSP establishment methodology provides between 128 and 256 bits of encryption strength	AES-KW
MACsec Key Derivation	KBKDF MAC	NIST SP 800-108 KDF used in the context of MAcsec to derive SSPs		KDF SP800-108 AES-CMAC AES-ECB AES-CBC
CASTs on boot	BC-Auth BC-UnAuth DigSig-SigGen DigSig-SigVer DRBG ENT-Cond KAS-135KDF KBKDF MAC SHA	List of algorithms for which Known Answer Tests (CASTs) have been implemented in the module and perform on each boot		AES-CBC HMAC DRBG HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-512 KAS-ECC-SSC Sp800-56Ar3 KAS-FFC-SSC Sp800-56Ar3 KDF SSH ECDSA

Name	Type	Description	Properties	Algorithms
				SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4) HMAC DRBG HMAC-SHA2-256 SHA2-512 AES-GCM AES-KW KDF SP800-108 SHA2-512 AES-CMAC

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

The module only supports testable RSA moduli/key sizes (2048, 3072 and 4096 bits) and thus the requirements per FIPS 140-3 IG C.F do not apply.

2.8 RBG and Entropy

Cert Number	Vendor Name
E104	Juniper Networks

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Junos OS Non-Physical Entropy Source	Non-Physical	Intel Xeon E3-1125v2	8 bits	0.83 bits	SHA2-512 (CAVP Cert. #A3403)

Table 11: Entropy Sources

2.9 Key Generation

The module implements two NIST SP 800-90Ar1 DRBGs and supports the following sections per NIST SP 800-133r2 (CKG): Sections 4, 5.1, 5.2 and 6.2.1.

2.10 Key Establishment

Per IG D.F:

The module implements full KAS (KAS-ECC-SSC, KAS-FFC-SSC per NIST SP 800-56Ar3 and KDF SSH per NIST SP 800-135r1; IG D.F Scenario 2 (path 2 option 2, separate testing of the SSC and SP800-135r1 KDF). The KAS1 and KAS2 in the SFI Table have been documented in accordance with this requirement:

KAS1: KAS (KAS-ECC-SSC Cert.#A4301 and CVL Cert. #A4301; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

KAS2: KAS (KAS-FFC-SSC Cert.#A4301 and CVL Cert. #A4301; SSP establishment methodology provides 112 bits of encryption strength)

The Approved Algorithm list includes the tested components (KAS-ECC-SSC, KAS-FFC-SSC and KDF SSH) as individual entries.

Per IG D.G:
The module supports the IETF SSH and MACsec protocols and thus implements key transport in the context of the protocols (per the KTS1 and KTS2 entries in the SFI table of the Security Policy).

The module implements the following approved KTS using approved AES modes:

AES CBC and CTR (KTS1): KTS (AES Cert. #A4301 and HMAC Cert. #A4301; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

AES KW (KTS2): KTS (AES Cert. #A4304; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

2.11 Industry Protocols

No parts of the SSH and MACsec protocols, other than the KDF SSH and the NIST SP 800-108 KDF for MACsec, have been tested by the CAVP or CMVP.

2.12 Additional Information

The module design corresponds to the security rules below. The term *shall* in this context specifically refers to a requirement for correct usage of the module in the Approved mode; all other statements indicate a security rule implemented by the module.

1. The module clears previous authentications on power cycle.
2. When the module has not been placed in a valid role, the operator does not have access to any cryptographic services.

3. Self-tests do not require any operator action.
4. Data output is inhibited during SSP generation, self-test execution, zeroisation, and error states.
5. Status information does not contain SSPs or sensitive data that if misused could lead to a compromise of the module.
6. There are no restrictions on which SSPs are zeroised by the zeroisation service.
7. The module does not support a maintenance interface or role.
8. The module does not output intermediate key values.
9. The module does not output plaintext CSPs.
10. The Crypto officer shall verify that the firmware image to be loaded on the module is a FIPS 140-3 validated image. If any non-validated firmware image is loaded the module will no longer be a validated module.
11. The Crypto Officer shall retain control of the module while zeroisation is in process.
12. MACsec protocol IV generation:
    - The AES GCM IV construction is performed internal to the module in compliance with IEEE 802.1AEand its amendments. The IV length is 96 bits (per SP 800-38D). The module ensures the IV is constructed deterministically per Section 8.2 in SP 800-38D and the MACsec standard IEEE 802.1AE as a result of concatenating the fixed field (SCI) and invocation field (PN).
    - The module can take on the role of Peer or Authenticator in reference to the MACsec protocol.
    - The module shall only be used with other FIPS 140-3 validated modules when supporting the MACsec protocol in the role of a Peer/Authenticator for providing the remaining functionalities.
    - Per FIPS 140-3 IG C.H Scenario 3, if the module loses power and then it is restored, then a new key shall be established for use with the AES GCM encryption/decryption processes.
    - The link between the Peer and Authenticator, used in the MACsec communication, shall be secure to prevent the possibility for an attacker to introduce foreign equipment into the local area network.
13. The module shall not be configured to use a radius server and the radius server capability shall be disabled.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Ethernet (Management port)	Data Input Data Output Control Input Status Output	LAN, Communications/remote management
Serial	Data Input Data Output	Console Serial Port

Physical Port	Logical Interface(s)	Data That Passes
	Control Input Status Output	
USB	Data Input Control Input	USB port, load Junos Image
Power	Power	Power connector, Power over Ethernet
Alarm LEDs	Status Output	Status indicator lighting
Reset Button	Control Input	Reset signal
PTP-capable connections	Data Input Data Output	SMB In/out (clock synchronization)
Online/Offline Button	Control Input	Online/Offline signal

Table 12: Ports and Interfaces

The module does not support control output.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Username and password over the console and SSH	• The module enforces 10-character passwords (at minimum) chosen from the 96 human readable ASCII characters; The maximum password length is 20-characters; Thus, the probability of a successful random attempt is 1/(96^10), which is less than 1/1,000,000 (million); • The module enforces a timed access mechanism as follows: For the first two failed attempts (assuming 0 time to process), no timed access is enforced; Upon the third attempt, the module enforces a 5-second delay; Each failed attempt thereafter results in an additional 5-second delay above the previous (e.g., 4th failed attempt = 10-second delay, 5th failed attempt = 15-second delay, 6th failed attempt = 20-second delay, 7th failed attempt	SHA2-512 (A4306)	1/(96^10)	9/(96^10)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	= 25-second delay); This leads to a maximum of 7 possible attempts in a one-minute period for each getty; 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/60 mins); this would be rounded down to 9 per minute, because there is no such thing as 0.6 attempts; The probability of a success with multiple consecutive attempts in a one-minute period is 9/(96^10), which is less than 1/100,000			
Username and ECDSA public key over SSH	• The module supports ECDSA (P-256, P-384, and P-521), which has a minimum equivalent computational resistance to attack of either 2^128, 2^192 or 2^256 depending on the curve; Thus, the probability of a successful random attempt is 1/(2^128), which is less than 1/1,000,000 (million) • Configurable SSH connection establishment rate limits the number of connection attempts, and thus failed authentication attempts in a one-minute period to a maximum of 15,000 attempts; The probability of a success with multiple consecutive attempts in a one-minute period is 15,000/(2^128), which is less than 1/100,000	ECDSA SigVer (FIPS186-4) (A4301)	1/(2^128)	15,000/(2^128)
Username and RSA public key over SSH	• The module supports RSA (2048, 3072, 4096 bits), which has a minimum equivalent computational resistance to attack of 2^112 (2048 bits); Thus, the probability of a successful random attempt is 1/ (2^112), which is less than 1/1,000,000 (million) • Configurable SSH	RSA SigVer (FIPS186-4) (A4301)	1/ (2^112)	15,000/(2^112)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	connection establishment rate limits the number of connection attempts, and thus failed authentication attempts in a one-minute period to a maximum of 15,000 attempts; The probability of a success with multiple consecutive attempts in a one-minute period is 15,000/(2^112), which is less than 1/100,000			

Table 13: Authentication Methods

The module enforces the separation of roles using identity-based operator authentication. The module implements two forms of identity-based authentication, username, and password over the console and SSH connections, as well as username and an ECDSA or RSA public key-based authentication over SSHv2.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Super-user	Identity	Crypto Officer (CO)	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Operator	Identity	User	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Read-only	Identity	User	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Root	Identity	Crypto Officer (CO)	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Unauthorised	Identity	User	Username and password over the console and SSH

Name	Type	Operator Type	Authentication Methods
			Username and ECDSA public key over SSH Username and RSA public key over SSH

Table 14: Roles

The module supports two roles: Crypto Officer (CO) and User. Root and Super-user correspond to the Crypto Officer role whereas Operator, Read-Only and Unauthorised operator types correspond to the User role. The module supports concurrent operators but does not support a maintenance role and/or bypass capability.

An operator assuming the Crypto Officer role configures and monitors the module via a console or SSH connection. As Root or Super-user, the Crypto Officer has permission to view and configure passwords and public keys within the module. The User role monitors the module via the console or SSH. The User role does not have the permission to modify the configuration.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure security (security relevant)	Security relevant configuration (SSH, authentication data)	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Commands (SSH configuration: set system services ssh root-login allow)	Traffic	DRBG DRBG2 Password Hash CKG	Root - SSH Private Host Key: G - User Password: W,E - CO Password: W,E - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - SSH Public Host Key: G - User Authenticatio

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						n Public Keys: W - CO Authentication Public Keys: W Super-user - SSH Private Host Key: G - User Password: W,E - CO Password: W,E - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - SSH Public Host Key: G - User Authentication Public Keys: W - CO Authentication Public Keys: W
Configure (non-security relevant)	Non-security relevant configuration	Global Approved Mode indicator “fips” at the CLI combined with success	Commands (miscellaneous commands e.g., for IP address configuration, routing	Traffic	Password Hash	Super-user - CO Password: E Root - CO Password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		ful completion of each service	protocols, etc.)			
Show status	Query the module status	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Command (show)	CLI output	Password Hash	Super-user - CO Password: E Root - CO Password: E Operator - User Password: E Read-only - User Password: E Unauthorised - User Password: E
Show status (LED)	LEDs on the module provide physical status output	LED(s) on the chassis turned on	N/A	LED	None	Super-user Operator Read-only Unauthorised Root Unauthenticated
Show module’s versioning information	Query the module’s versioning information	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Command (show version)	CLI output	Password Hash	Super-user - CO Password: E Operator - User Password: E Read-only - User Password: E Unauthorised - User Password: E Root - CO Password: E
Zeroise (Perform	Destroy all SSPs	Global Approved Mode	Command (request vmhost	N/A	Password Hash	Super-user - SSH Private Host

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
zeroisation)		indicator “fips” at the CLI combined with successful completion of each service	zeroise no-forwarding)			Key: Z - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - User Password: Z - CO Password: E,Z - HMAC_DRBG V value: Z - HMAC_DRBG Key value: Z - HMAC_DRBG entropy input: Z - HMAC_DRBG seed: Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: Z - SSH Public Host Key: Z - User Authentication Public Keys: Z - CO Authentication Public Keys: Z - JuniperRoot CA: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- PackageCA: Z - SSH ECDH Public Key: Z - SSH DH Public Key: Z - SSH ECDH Client Public Key: Z - SSH DH Client Public Key: Z - MACsec PSK: Z - MACsec SAK: Z - MACsec KEK: Z - MACsec ICK: Z Root - SSH Private Host Key: Z - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - User Password: Z - CO Password: E,Z - HMAC_DRBG V value: Z - HMAC_DRBG Key value: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						HMAC_DRBG entropy input: Z - HMAC_DRBG seed: Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: Z - SSH Public Host Key: Z - User Authentication Public Keys: Z - CO Authentication Public Keys: Z - JuniperRoot CA: Z - PackageCA: Z - SSH ECDH Public Key: Z - SSH DH Public Key: Z - SSH ECDH Client Public Key: Z - SSH DH Client Public Key: Z - MACsec PSK: Z - MACsec SAK: Z - MACsec KEK: Z - MACsec ICK: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform approved security functions (SSH connection)	Initiate SSH connection for SSH monitoring and control (CLI)	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Authentication data (Username and password/public-key based authentication)	SSH session	KAS1 KAS2 KTS1 ECDSA SigVer2 DRBG DRBG2 Entropy Souce ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen RSA SigVer Password Hash CKG	Super-user - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: G - SSH DH Public Key: G,E,Z - SSH ECDH Public Key: G,E,Z - CO Password: E - CO Authentication Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Root - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Public Key: G,E,Z - CO Password: E - CO Authentication Public Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Operator - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public Keys: E - HMAC_DRBG Key value: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Read-only - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Unauthorised - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG entropy input: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public Keys: E - HMAC_DRBG Key value: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z
Console Access	Console monitoring and control (CLI)	Global Approved Mode indicator “fips” at the CLI combined with successful completion of	Username, password (set system login user <username> class <crypto-officer/user class> operator authentication plaintext-password)	N/A	Password Hash	Super-user - CO Password: E Operator - CO Password: E Read-only - User Password: E Unauthorised - User Password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		each service				Root - CO Password: E
Perform self-tests (remote reset)	Software initiated reset, performs self-tests on demand via SSH	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Control input/reset signal (request vmhost reboot)	N/A	KAS1 KAS2 KTS1 DRBG DRBG2 Entropy Souce ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen Password Hash CKG CASTs on boot	Super-user - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - HMAC_DRBG Key value: G,Z - HMAC_DRBG V value: G,Z - HMAC_DRBG entropy input: G,Z - HMAC_DRBG seed: G,Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: G,E,Z - SSH ECDH Public Key: G,E - SSH DH Public Key: G,E - CO Password: E - Firmware Integrity Key: E - SSH Private Host Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E Root - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - HMAC_DRBG Key value: G,Z - HMAC_DRBG V value: G,Z - HMAC_DRBG entropy input: G,Z - HMAC_DRBG seed: G,Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: G,E,Z - SSH ECDH Public Key: G,E - SSH DH Public Key: G,E - CO

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Password: E - Firmware Integrity Key: E - SSH Private Host Key: E - SSH Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E
Perform self-tests (local reset)	Hardware reset or power cycle	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Control input/reset signal	N/A	CASTs on boot	Super-user - Firmware Integrity Key: E Root - Firmware Integrity Key: E Operator - Firmware Integrity Key: E Read-only - Firmware Integrity Key: E Unauthorised - Firmware Integrity Key: E Unauthenticated - Firmware Integrity Key: E
Load Image	Verification and loading of a validated firmware image into	Global Approved Mode indicator “fips” at the CLI	Image, commands	N/A	ECDSA SigVer Password Hash	Super-user - CO Password: E - Firmware Integrity Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	the router/switch	combined with successful completion of each service				- JuniperRootCA: E - PackageCA: E Root - CO Password: E - Firmware Integrity Key: E - JuniperRootCA: E - PackageCA: E
Perform approved security functions (MACsec connection)	Initiate MACsec connection	Global Approved Mode indicator “fips” at the CLI combined with successful completion of each service	Commands (set security macsec connectivity-association connectivity-association-name; set security macsec connectivity-association connectivity-association-name pre-shared key)	MACsec session	CKG MACsec Encryption/Decryption KTS2 MACsec Key Derivation	Root - MACsec PSK: W,E - MACsec SAK: G,R,E - MACsec KEK: G,E - MACsec ICK: G,E Super-user - MACsec PSK: W,E - MACsec SAK: G,R,E - MACsec KEK: G,E - MACsec ICK: G,E

Table 15: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Configure security (security relevant)	Security relevant configuration	RSA with key size less than 2048 ECDSA with ed25519 curve	Root, Super-user

Name	Description	Algorithms	Role
		EC Diffie-Hellman with ed25519 curve ARCFOUR Blowfish CAST DSA (SignGen, SigVer, non-compliant) HMAC-MD5 HMAC-RIPEMD160 UMAC	
Configure (non-security relevant)	Non-security relevant configuration	None	Root, Super-user
Show status	Query the module status	None	Root, Super-user, Operator, Read-Only, Unauthorized
Show status (LED)	LEDs on the module provide physical status output	None	Root, Super-user, Operator, Read-Only, Unauthorized, Unauthenticated
Show module's versioning information	Query the module's versioning information	None	Root, Super-user, Operator, Read-Only, Unauthorized
Zeroise (Perform zeroisation)	Destroy all SSPs	None	Root, Super-user
Perform approved security functions (SSH connection)	Initiate SSH connection for SSH monitoring and control (CLI)	RSA with key size less than 2048 ECDSA with ed25519 curve EC Diffie-Hellman with ed25519 curve ARCFOUR Blowfish CAST DSA (SignGen, SigVer, non-compliant) HMAC-MD5 HMAC-RIPEMD160 UMAC	Root, Super-user, Operator, Read-Only, Unauthorized
Console Access	Console monitoring and control (CLI)	None	Root, Super-user, Operator, Read-Only, Unauthorized

Name	Description	Algorithms	Role
Perform self-tests (remote reset)	Software initiated reset, performs self-tests on demand	None	Root, Super-user, Operator, Read-Only, Unauthorized
Perform self-tests (local reset)	Hardware reset or power cycle	None	Root, Super-user, Operator, Read-Only, Unauthorized, Unauthenticated
Load Image	Verification and loading of a validated firmware image into the router/switch	None	Root, Super-user
Perform approved security functions (MACsec connection)	Initiate MACsec connection	None	Root, Super-user

Table 16: Non-Approved Services

4.5 External Software/Firmware Loaded

The module supports loading of firmware from an external source (a complete image replacement) and a firmware load test using ECDSA P-256 with SHA2-256 (CAVP Cert. #A4301) is performed in support of the load.

4.6 Cryptographic Output Actions and Status

The module supports self-initiated cryptographic output in the context of the MACsec protocol and three independent configurations are required serving as three independent internal actions (two actions required at minimum):

- set security macsec connectivity-association <name> cipher-suite
- set interfaces <name> connectivity-association <name>
- set interfaces <name> unit 0 family inet address <ip address>

The following “show” commands indicate the status of the MACsec service:

- show security macsec connections
- show security mka sessions
- show security mka statistics

5 Software/Firmware Security

5.1 Integrity Techniques

The module performs the firmware integrity check using ECDSA P-256 with SHA2-256 (CAVP Cert. #A4301). The ECDSA P-256 public key used for signature verification is a non-SSP and stored persistently across reboots in the module’s Non-Volatile RAM (NVRAM) and is exempt from zeroisation.

5.2 Initiate on Demand

The operator can initiate the integrity test on demand by rebooting the module.

5.3 Additional Information

The module firmware image is delivered in the form of a pre-compiled tarball (.tgz).

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

The module contains a limited operational environment since it supports loading of firmware from an external source. The Junos OS 22.4R2.8 operating system is contained within the module, i.e., the tested configurations listed in the Tested Module Identification – Hardware in this document.

6.2 Configuration Settings and Restrictions

Security rules and restrictions for configuration of the operational environment have been specified in Sections 2.12 and 11.1 of this document.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
N/A	N/A	N/A

Table 17: Mechanisms and Actions Required

The module's physical embodiment is that of a multi-chip standalone meeting 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. The module enclosure is made of production grade materials. 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. No actions are required by the operator to ensure that physical security is maintained.

8 Non-Invasive Security

8.1 Mitigation Techniques

The module does not implement any non-invasive security mitigations and thus the requirements per this section do not apply to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
NVRAM	Non-Volatile Random Access Memory	Static
RAM	Random Access Memory	Dynamic

Table 18: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Entered over SSH - NVRAM	External endpoint	NVRAM	Encrypted	Automated	Electronic	KTS1
Loaded at manufacture	External endpoint	NVRAM	Plaintext	N/A	N/A	
Entered through the CLI via console connection - NVRAM	External endpoint	NVRAM	Plaintext	Manual	Direct	
Output encrypted with MAcsec KEK	RAM	External endpoint (MACsec peer)	Encrypted	Automated	Electronic	KTS2
Input during SSH negotiation	External endpoint	RAM	Plaintext	Automated	Electronic	
Output during SSH negotiation (host key)	NVRAM	External endpoint	Plaintext	Automated	Electronic	
Output during SSH negotiation (Key Agreement public key)	RAM	External endpoint	Plaintext	Automated	Electronic	

Table 19: SSP Input-Output Methods

The module is complaint with FIPS 140-3 IG 9.5.A MD/DE and AD/EE for SSPs entered via the module's CLI via a direct connection to its serial/console port and for SSPs entered/output/established via SSH/MACsec respectively.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroisation command	Command used to zeroise the module: request vmhost zeroize no-forwarding	Used to provide zeroisation as a service	Operator initiated
Power-cycle	Power cycling the module to zeroise temporary SSPs	Power cycling the module to zeroise temporary SSPs	Operator initiated
Session termination	Termination of SSH sessions automatically zeroises temporary SSPs used as part of the session	Termination of SSH sessions automatically zeroises temporary SSPs used as part of the session	Module initiated
Not zeroised	PSP not zeroised since it cannot be modified due to being inaccessible in the filesystem	PSP not zeroised since it cannot be modified due to being inaccessible in the filesystem	N/A
Derivation of SSH session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of SSH session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of SSH session key	Module initiated

Table 20: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH Private Host Key	Host key generated, used for authentication and encryption in the context of SSH	P-256 for ECDSA, 2048 bits for RSA - 128 bits for ECDSA, 112 bits for RSA	Private Host Key - CSP	DRBG2 ECDSA KeyGen RSA KeyGen		KAS1 KAS2
SSH ECDH Private Key	Ephemeral EC Diffie-Hellman private key used in SSH	KAS-ECC-SSC P-256, P-	ECDH Private Key - CSP	DRBG2 ECDSA KeyGen2		KAS1

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		384, P-512 - 128 bits, 192 bits, 256 bits				
SSH DH Private Key	Ephemeral Diffie-Hellman private key used in SSH	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	DH Private Key - CSP	DRBG2		KAS2
SSH Session Key	SSH Session Key	128 bits, 192 bits, 256 bits - 128 bits, 192 bits, 256 bits	Session Key - CSP	CKG	KAS1 KAS2	
User Password	Passwords used to authenticate users to the module	10-20 characters - 1/(96^10) per attempt, 9/(96^10) per minute	User Password - CSP			
CO Password	Passwords used to authenticate COs to the module	10-20 characters - 1/(96^10) per attempt, 9/(96^10) per minute	CO Password - CSP			
HMAC_DRBG V value	A critical value of the internal state of DRBG	256 bits - 256 bits	Internal state of the DRBG - CSP	DRBG DRBG2		DRBG DRBG2
HMAC_DRBG Key value	A critical value of the internal state of DRBG	440 bits - 440 bits	Internal state of the DRBG - CSP	DRBG DRBG2		DRBG DRBG2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC_DRBG entropy input	Entropy input to the HMAC_DRBG	512 bits - 448 bits	Entropy input to the HMAC_DRBG - CSP	Entropy Souce		
HMAC_DRBG seed	Seed provided to the HMAC_DRBG	512 bits - 440 bits	Seed provided to the HMAC_DRBG - CSP	DRBG DRBG2		DRBG DRBG2
ECDH Shared Secret	Used in EC Diffie-Hellman (ECDH) exchange	P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Shared secret - CSP		KAS1	
DH Shared Secret	Used in Diffie-Hellman (DH) exchange	2048 bits - 112 bits	Shared secret - CSP		KAS2	
HMAC Key	MAC key	128 bits and 256 bits - 128 bits and 256 bits	MAC key - CSP		KAS1 KAS2	
SSH Public Host Key	Host key generated, used to identify the host. Also paired with the private key for authentication and encryption in the context of SSH	P-256 for ECDSA and 2048 bits for RSA - 128 bits for ECDSA, 112 bits for RSA	Public key - PSP	DRBG2 ECDSA KeyGen RSA KeyGen		
User Authentication Public Keys	Used to authenticate users to the module	P-256, P-384, P-521 for ECDSA and 2048, 3072 and 4096 bits for	Public key - PSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		RSA - 128, 192, 256 bits for ECDSA, 112, 192 and 256 bits for RSA				
CO Authentication Public Keys	Used to authenticate the CO to the module	P-256, P-384, P-521 for ECDSA and 2048, 3072 and 4096 bits for RSA - 128, 192, 256 bits for ECDSA, 112, 192 and 256 bits for RSA	Public key - PSP			
JuniperRoot CA	ECDSA prime256v1 X.509 V3 Certificate Used to verify the validity of the PackagCA	ECDSA P-256 - 128 bits	Public key certificate - Neither			
PackageCA	ECDSA prime256v1 X.509 V3 Certificate Certificate that holds the public key for the signing key used to generate all the signatures used on the packages and signature lists	ECDSA P-256 - 128 bits	Public key certificate - Neither			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH ECDH Public Key	Ephemeral EC Diffie-Hellman public key used in SSH	KAS-ECC-SSC P-256, P-384, P-512 - 128 bits, 192 bits, 256 bits for KAS-ECC-SSC	Public key - PSP	DRBG2 ECDSA KeyGen2		
SSH DH Public Key	Ephemeral Diffie-Hellman public key used in SSH	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	Public key - PSP	DRBG2		
Firmware Integrity Key	Public key used to perform the firmware integrity test on each boot and authenticate firmware loaded from an external source	ECDSA P-256 - 128 bits	Public key - Neither			
MACsec PSK	Credential used for device-to-device authentication, consists of the CAK (pre-shared key) and CKN (identifier for the pre-shared key)	128, 256 bits - 128, 256 bits	Symmetric key - CSP			
MACsec SAK	Security Association Key used for creating Security Associations for encryption/decryption of MACsec traffic	128, 256 bits - 128, 256 bits	Symmetric key - CSP	MACsec Key Derivation		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
MACsec KEK	Used to transmit SAKs to other members of a MACsec connectivity association	128, 256 bits - 128, 256 bits	Symmetric key - CSP	MACsec Key Derivation		
MACsec ICK	Used to verify the integrity and authenticity of MACsec protocol data units	128, 256 bits - 128, 256 bits	Symmetric key - CSP	MACsec Key Derivation		
SSH ECDH Client Public Key	Ephemeral EC Diffie-Hellman public key used in SSH (sent by the client to the module acting as the server)	KAS-ECC-SSC P-256, P-384, P-512 - 128 bits, 192 bits, 256 bits for KAS-ECC-SSC	Public key - PSP			
SSH DH Client Public Key	Ephemeral Diffie-Hellman public key used in SSH (sent by the client to the module acting as the server)	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	Public key - PSP			

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH Private Host Key		NVRAM:Plaintext		Zeroisation command	
SSH ECDH Private Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Private Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Session termination	
SSH Session Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
User Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated NVRAM:Obfuscated		Zeroisation command	
CO Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated NVRAM:Obfuscated		Zeroisation command	
HMAC_DRBG V value		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG Key value		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG entropy input		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG seed		RAM:Plaintext	Until power-cycle	Power-cycle	
ECDH Shared Secret		RAM:Plaintext	Until SSH session key derivation	Zeroisation command Power-cycle Derivation of SSH session key	
DH Shared Secret		RAM:Plaintext	Until SSH session key derivation	Zeroisation command Power-cycle Derivation	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				of SSH session key	
HMAC Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH Public Host Key	Output during SSH negotiation (host key)	NVRAM:Plaintext		Zeroisation command	
User Authentication Public Keys	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Zeroisation command	
CO Authentication Public Keys	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Zeroisation command	
JuniperRootCA	Loaded at manufacture	NVRAM:Plaintext		Not zeroised	
PackageCA	Loaded at manufacture	NVRAM:Plaintext		Not zeroised	
SSH ECDH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
Firmware Integrity Key	Loaded at manufacture	NVRAM:Plaintext		Not zeroised	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
MACsec PSK	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Zeroisation command	
MACsec SAK	Output encrypted with MAcsec KEK	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
MACsec KEK		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
MACsec ICK		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH ECDH Client Public Key	Input during SSH negotiation	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Client Public Key	Input during SSH negotiation	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	

Table 22: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware Integrity Test	Using ECDSA P-256 with SHA2-256	KAT	SW/FW Integrity	FIPS Self-tests Passed	Verify

Table 23: Pre-Operational Self-Tests

The module is complaint with FIPS 140-3 IG 10.2.A in that it performs a self-test, a Known Answer Test (KAT) for the ECDSA P-256 (with SHA2-256) algorithm used in the firmware integrity test on each boot prior to executing the firmware integrity test.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG (A4303)	Prediction Resistance: Yes Supports Reseed Capabilities: Mode: SHA2-256 Entropy Input: 256 Nonce: 128 Personalization String Length: 0-256 Increment 8 Additional Input: 8-256 Increment 8 Returned Bits: 1024	KAT	CAST	NIST 800-90 HMAC DRBG Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-256 (A4303)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
AES-CBC (A4301)	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A4301)	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A4301)	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer	Encrypt	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Test : Passed		
AES-CBC (A4301)	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
AES-CBC (A4301)	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
AES-CBC (A4301)	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
HMAC DRBG (A4301)	Mode: SHA2-256, Entropy Input: 256 , Nonce: 128, Personalization String Length: 0-256 , Increment 8 , Additional Input: 8-256 Increment 8 , Returned Bits: 1024	KAT	CAST	NIST 800-90 HMAC DRBG Known Answer Test : Passed	N/A	During boot
HMAC-SHA-1 (A4301)	Key Length: 160 bits	KAT	CAST	HMAC-SHA-1 Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-256 (A4301)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-	Key Length: 512 bits	KAT	CAST	HMAC-SHA2-512 Known	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
512 (A4301)				Answer Test : Passed		
KAS-ECC-SSC Sp800-56Ar3 (A4301)	Domain Parameter Generation Methods: P-256	KAT	CAST	KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed	N/A	During boot
KAS-ECC-SSC Sp800-56Ar3 (A4301)	Domain Parameter Generation Methods: P-384	KAT	CAST	KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed	N/A	During boot
KAS-FFC-SSC Sp800-56Ar3 (A4301)	Domain Parameter Generation Methods: MODP-2048	KAT	CAST	KAS-FFC-EPHEM-NOKC Known Answer Test: Passed	N/A	During boot
KDF SSH (A4301)	Cipher: AES-128, AES-192, AES-256 ; Hash Algorithm: SHA-1, SHA2-256, SHA2-512	KAT	CAST	KDF-SSH-SHA2-256 Known Answer Test: Passed	N/A	During boot
RSA SigGen (FIPS186-4) (A4301)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-SIGN Known Answer Test: Passed	Sign	During boot
RSA SigVer (FIPS186-4) (A4301)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-VERIFY Known Answer Test: Passed	Verify	During boot
ECDSA SigGen	Curve: P-256 Hash	KAT	CAST	ECDSA-SIGN	Sign	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A4301)	Algorithm: SHA2-256			Known Answer Test: Passed		
ECDSA SigVer (FIPS186-4) (A4301)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	CAST	ECDSA-VERIFY Known Answer Test: Passed	Verify	During boot
SHA2-512 (A4306)	SHA2-512	KAT	CAST	SHA-2-512 Known Answer Test: Passed	N/A	During boot
Entropy test	NIST SP 800-90B Repetitive Count Test	RCT	CAST	pass	Cutoff value C = 21	During boot and continually
Entropy test	NIST SP 800-90B Adapative Proportion Test	APT	CAST	pass	W = 512; Cutoff value C = 311	During boot and continually
ECDSA KeyGen (FIPS186-4) (A4301)	Curve: P-256 Hash Algorithm: SHA2-256	PCT	PCT	0	Key pair generated for signature generation/verification in the context of SSHv2 protocol	On key generation
ECDSA KeyGen (FIPS186-4) (A4301)	Curve: P-256 Hash Algorithm: SHA2-256	PCT	PCT	0	Key pair generated for SSP agreement in the context of SSHv2 protocol	On key generation
KAS-FFC-SSC Sp800-56Ar3 (A4301)	Capabilities: Domain Parameter: MODP2048	PCT	PCT	0	Key pair generated for SSP agreement in the context of SSHv2 protocol	On key generation
RSA KeyGen (FIPS186-4) (A4301)	Modulus: 2048 Hash SHA2-256	PCT	PCT	0	Key pair generated for signature generation/verification in the context of SSHv2 protocol	On key generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-KW (A4304)	Key Length: 128 bits	KAT	CAST	AES-KEYWRAP Known Answer Test: Passed	Encrypt	During boot
AES-KW (A4304)	Key Length: 128 bits	KAT	CAST	AES-KEYWRAP Known Answer Test: Passed	Decrypt	During boot
KDF SP800-108 (A4304)	Mode: Counter	KAT	CAST	KBKDF Known Answer Test:Passed	N/A	During boot
AES-GCM (AES 4369)	Key Length: 256 bits	KAT	CAST	0	Encrypt	During boot
AES-GCM (AES 4369)	Key Length: 256 bits	KAT	CAST	0	Decrypt	During boot
AES-CMAC (A4304)	Key Length: 128 bits	KAT	CAST	AES128-CMAC Known Answer Test: Passed	Encrypt	During boot
AES-CMAC (A4304)	Key Length: 128 bits	KAT	CAST	AES128-CMAC Known Answer Test: Passed	Decrypt	During boot
AES-CMAC (A4304)	Key Length: 256 bits	KAT	CAST	AES256-CMAC Known Answer Test: Passed	Encrypt	During boot
AES-CMAC (A4304)	Key Length: 256 bits	KAT	CAST	AES256-CMAC Known Answer	Decrypt	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Test: Passed		
ECDSA SigVer (FIPS186-4) (A4301)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	SW/FW Load	Host OS upgrade staged. Reboot the system to complete installation!	Verify	On loading of firmware from an external source
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's console (serial) interface	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's console (serial) interface	Manual Entry	Command prompt with "fips" string provided post completion of the test	N/A	On configuration of operator passwords and MACsec PSK

Table 24: Conditional Self-Tests

Cryptographic Algorithm Self-tests (CASTs) are performed on each boot of the module. Other conditional self-tests are performed by the module when the corresponding condition is met. The pairwise consistency tests are performed on key pair generation for use in signature generation/verification (ECDSA and/or RSA tests) and/or for use in KAS-ECC-SSC or KAS-FFC-SSC SSP agreement (ECDSA and FFC tests respectively). The firmware load test is performed when a firmware image is loaded onto the module from an external source.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware Integrity Test	KAT	SW/FW Integrity	On Demand	Manually via a reboot

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A4303)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A4303)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A4301)	KAT	CAST	On Demand	Manually via a reboot
HMAC DRBG (A4301)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA-1 (A4301)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A4301)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-512 (A4301)	KAT	CAST	On Demand	Manually via a reboot
KAS-ECC-SSC Sp800-56Ar3 (A4301)	KAT	CAST	On Demand	Manually via a reboot
KAS-ECC-SSC Sp800-56Ar3 (A4301)	KAT	CAST	On Demand	Manually via a reboot
KAS-FFC-SSC Sp800-56Ar3 (A4301)	KAT	CAST	On Demand	Manually via a reboot
KDF SSH (A4301)	KAT	CAST	On Demand	Manually via a reboot
RSA SigGen (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually via a reboot
RSA SigVer (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually via a reboot
ECDSA SigGen (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually via a reboot
ECDSA SigVer (FIPS186-4) (A4301)	KAT	CAST	On Demand	Manually via a reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-512 (A4306)	KAT	CAST	On Demand	Manually via a reboot
Entropy test	RCT	CAST	On Demand	Manually via a reboot
Entropy test	APT	CAST	On Demand	Manually via a reboot
ECDSA KeyGen (FIPS186-4) (A4301)	PCT	PCT	On Demand	Manually via a reboot
ECDSA KeyGen (FIPS186-4) (A4301)	PCT	PCT	On Demand	Manually via a reboot
KAS-FFC-SSC Sp800-56Ar3 (A4301)	PCT	PCT	On Demand	Manually via a reboot
RSA KeyGen (FIPS186-4) (A4301)	PCT	PCT	On Demand	Manually via a reboot
AES-KW (A4304)	KAT	CAST	On Demand	Manually via a reboot
AES-KW (A4304)	KAT	CAST	On Demand	Manually via a reboot
KDF SP800-108 (A4304)	KAT	CAST	On Demand	Manually via a reboot
AES-GCM (AES 4369)	KAT	CAST	On Demand	Manually via a reboot
AES-GCM (AES 4369)	KAT	CAST	On Demand	Manually via a reboot
AES-CMAC (A4304)	KAT	CAST	On Demand	Manually via a reboot
AES-CMAC (A4304)	KAT	CAST	On Demand	Manually via a reboot
AES-CMAC (A4304)	KAT	CAST	On Demand	Manually via a reboot
AES-CMAC (A4304)	KAT	CAST	On Demand	Manually via a reboot
ECDSA SigVer (FIPS186-4) (A4301)	KAT	SW/FW Load	On Demand	Manually via loading of firmware from an external source
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's	Manual Entry	On Demand	Manually via configuration of operator passwords and MACsec PSK

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
	console (serial) interface			

Table 26: Conditional Periodic Information

The pre-operational firmware integrity test as well as all CASTs must be completed successfully prior to any other use of cryptography by the module in the Approved mode of operation. These tests can also be performed periodically by rebooting the module.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard Error state	If the pre-operation firmware integrity test, if any of the CASTs or pair-wise consistency tests fail, then the module returns an error indicator, inhibits all data output and enters the hard error state	If the pre-operational firmware integrity test, fails, if any of the CASTs fail or if a pairwise consistency test fails	N/A	"FIPS error: self-test failure" for firmware integrity failure, "FIPS error 1: <name of the algorithm> Known Answer Test: Failed" for CAST failure and -1 for pair-wise consistency test failure
Soft Error state	•In case of a firmware load test failure, the module rejects the firmware, returns an error indicator and enters the soft error state •In the event of an APT or RCT health test failure, output from the entropy source is inhibited, all entropy accumulated in the conditioning context is discarded and the start-up health-tests are performed again	If the firmware load test fails If the APT or RCT test fails	N/A for firmware load test failure; In case of APT and/or RCT failures, new data continues to be tested by the health tests, and once both health tests indicate a “pass”, the entropy source again outputs data	"Validation Error" for the firmware load test failure; entropy data discarded in case of APT/RCT failure

Table 27: Error States

If the pre-operation firmware integrity test or if any of the CASTs fail, then the module returns the error indicator “FIPS error: self-test failure”, inhibits all data output and enters the hard error state.

If the conditional self-tests fail, the module enters the soft error state, i.e., it rejects the generated keypair/loaded image, returns an error indicator and resumes normal operation.

10.5 Operator Initiation of Self-Tests

Each time the module is powered up it tests that all the cryptographic algorithms operate correctly, and that sensitive data have not been damaged. Pre-operational as well as Conditional Cryptographic Algorithm Self-tests (CAST) are performed on each power up/boot of the module and on demand by power cycling the module (Perform self-tests (remote reset) service).

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Crypto Officer must follow the procedures defined below for secure installation, initialization, startup and operation of the module.

Crypto Officer Guidance
The Crypto Officer must check to verify the firmware image being loaded on the module is the FIPS 140-3 validated version/image. If the image is the FIPS 140-3 validated image, then proceed with installation of the image.

Installing The Firmware Image
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.

Connect to the console port on the device from your management device and log in to the Junos OS CLI. Copy the firmware package to the device to the /var/tmp/ directory. Install the new package on the device using the following command: operator> request vmhost software add /var/tmp/<package>.tgz.

NOTE: If you need to terminate the installation, do not reboot your device; instead, finish the installation and then issue the request system software delete package.tgz command, where package.tgz is, for example, junos-vmhost-install-ptx-x86-64-22.4R2.8.tgz. This is your last chance to stop the installation.

Reboot the device to complete the load and start the installation:

operator> request vmhost reboot

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

Also install the built-in fips-mode.tgz package needed for enabling the Approved-mode and the jpfe-fips package needed for execution of the CASTs. Please note that this is a one-time installation after which the module remains in the Approved mode once enabled and automatically executes the CASTs on each boot without requiring any operator or external intervention. The following are the commands used for installing these packages:
operator >request system software add optional://fips-mode.tgz
operator >request system software add optional://jpfe-fips.tgz

Enabling Approved Mode of Operation
The Crypto Officer is responsible for initializing the module in the Approved mode of operation. The Approved mode of operation is not automatically enabled. The Crypto Officer shall place the module in the Approved mode by first zeroising it to ensure no SSPs are present. Next, the cryptographic officer shall follow the steps found in the Junos OS FIPS Evaluated Configuration Guide for PTX Series, Release 22.4R2 document Chapter 2 to place the module into an Approved mode of operation. The steps from the aforementioned document have been reiterated below.

To enable the Approved mode in Junos OS on the module:

1. Zeroise the module using the “request vmhost zeroize” command. Once the module comes up in the “amnesiac mode” post zeroisation, connect to it using the console port with username “root”, enter the configuration mode and configure the root-authentication password (i.e., Crypto Officer credentials) as follows:

   root@device> edit
   Entering configuration mode

   [edit]
   root@device# set system root-authentication plain-text-
   password New password:
   Retype new password:

   [edit]
   root@device# commit
   configuration check succeeds
   commit complete

2. Enable Approved mode on the device by setting the Approved level to 1, and verify the level:

   [edit]
   root@device# set system fips level 1

   [edit]

   root@device# show system fips level
   level 1;

4. Commit the configuration

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

5. Reboot the device:

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

During the reboot, the device runs the pre-operational firmware integrity test and all CASTs. It returns a login prompt as follows:

```
root@device:fips>
```

6. After the reboot has completed, log in and use the show version command to verify the firmware version is the validated version:

```
root@device:fips > show version
```

Placing the Module in the Non-Approved Mode of Operation

As Crypto Officer, the operator needs to disable the Approved mode of operation on the device to return it to the non-Approved mode of operation. To disable the Approved mode on the device, the module must be zeroised (step 1 defined above).

11.2 Administrator Guidance

For further information and for the Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for PTX, Release 22.4R2 document.

11.3 Non-Administrator Guidance

For further information and for the Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for PTX, Release 22.4R2 document.

11.4 Maintenance Requirements

No other maintenance requirements apply for operation of the module in the Approved/non-Approved modes as defined above.

11.5 End of Life

The module can be securely sanitized at the end of its lifetime by zeroising it.

12 Mitigation of Other Attacks

12.1 Attack List

The module does not implement any mitigation of other attacks and thus the requirements per this section do not apply to the module.