Juniper Networks, Inc.

Juniper Networks vSRX 3.0 Virtual Firewall

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

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.

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 may be freely reproduced and distributed in its entirety without modification.

This non-proprietary Cryptographic Module Security Policy for the Juniper Networks vSRX 3.0 Virtual Firewall 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. It contains specification of the security rules under which the cryptographic module operates. The module will be running in approved mode of operation, when it is executing the Junos OS 22.2R2-S2.3 software version.

The Juniper Networks vSRX 3.0 Virtual Firewall may also be referred to as the “module” in this document.

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
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1

Section	Title	Security Level
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 vSRX 3.0 Virtual Firewall cryptographic module is comprised of the Junos OS 22.2R2-S2.3 software. The Juniper Networks vSRX 3.0 Virtual Firewall is a secure firewall that provides essential capabilities to connect, secure, and manage work force locations sized from handfuls to hundreds of users. By consolidating fast, highly available switching, routing, security, and applications capabilities in a single device, enterprises can economically deliver new services, safe connectivity, and a satisfying end user experience.

Module Type: Software

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The cryptographic boundary of the module is depicted in Figure 1 below. The physical perimeter is defined as the outer edge of the hardware platform (server) on which the hypervisor and Juniper Networks vSRX 3.0 Virtual Firewall are installed. The cryptographic boundary is the Juniper vSRX 3.0 Virtual Firewall which is comprised of the Junos OS 22.2R2-S2.3 software.

Tested Operational Environment’s Physical Perimeter (TOEPP):

The Tested Operational Environment’s Physical Perimeter (TOEPP) is the hardware platform on which it executes.

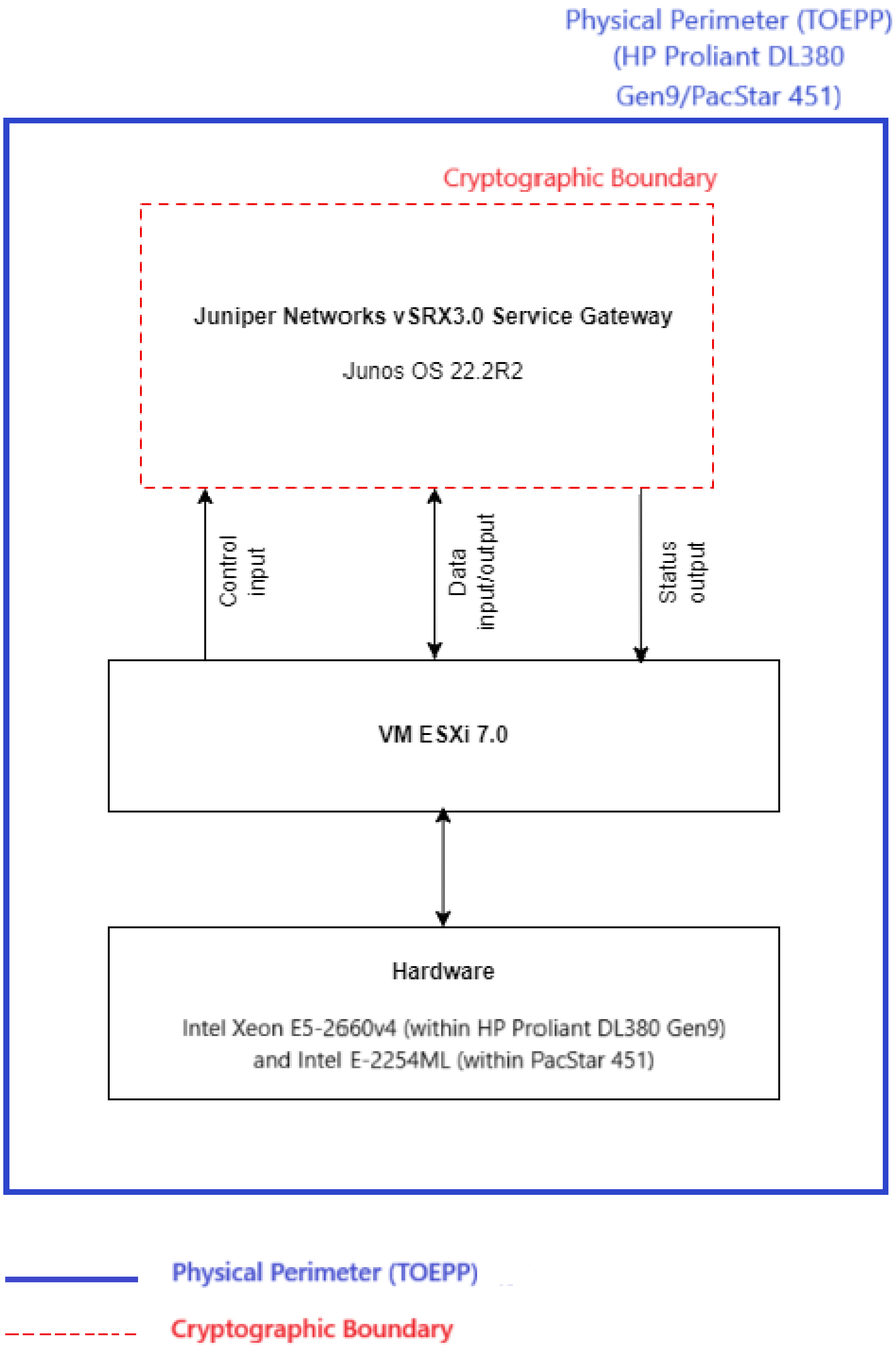
Physical Perimeter (TOEPP) (HP Proliant DL380 Gen9/PacStar 451)
Cryptographic Boundary
Juniper Networks vSRX3.0 Service Gateway
Junos OS 22.2R2
Control input
Data input/output
Status output
VM ESXi 7.0
Hardware
Intel Xeon E5-2660v4 (within HP Proliant DL380 Gen9) and Intel E-2254ML (within PacStar 451)
Physical Perimeter (TOEPP)
Cryptographic Boundary

Figure 1 – Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
junos-vsrx3-x86-64-22.2R2-S2.3.scsi.ova	Junos OS 22.2R2-S2.3	N/A	ECDSA P-256 with SHA2-256

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Junos OS 22.2R2-S2.3	HP ProLiant DL380 Gen9 Server	Intel Xeon E5-2660 v4	No	VMware ESXi 7.0	Junos OS 22.2R2-S2.3
Junos OS 22.2R2-S2.3	PacStar 451 Server	Intel Xeon E-2254ML	No	VMware ESXi 7.0	Junos OS 22.2R2-S2.3

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

No vendor-affirmed operational environments have been claimed.

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. crypto-officer:fips#); * The "show version" command will allow the	Approved	global indicator (string 'fips' included in the command prompt)

Mode Name	Description	Type	Status Indicator
	Crypto Officer to verify that the validated software version is running on the module; * The Crypto Officer can also use the "show system fips" command under configuration mode (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))		
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 * The module must be zeroised to transition from the Approved mode to the non-Approved mode	Non-Approved	global indicator (implicit indicator based on exclusion of string 'fips' from the command prompt)

Table 4: Modes List and Description

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. When switching from the non-Approved to the Approved mode, post zeroisation, the instructions in Section 11.1 Enabling the Approved Mode of Operation, must be followed.

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	A3339	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A3342	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A3343	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CTR	A3342	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
ECDSA KeyGen (FIPS186-5)	A3342	Curve - P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A3342	Curve - P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A3342	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A3342	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC DRBG	A3335	Prediction Resistance - Yes Mode - SHA2-256	SP 800-90A Rev. 1
HMAC-SHA-1	A3342	Key Length - Key Length: 160	FIPS 198-1
HMAC-SHA2-256	A3335	Key Length - Key Length: 160, 256	FIPS 198-1
HMAC-SHA2-256	A3339	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A3342	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A3343	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-512	A3342	Key Length - Key Length: 512	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A3342	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	A3342	Domain Parameter Generation Methods - FC, MODP-2048 Scheme - dhEphem - KAS Role - initiator	SP 800-56A Rev. 3
KDF IKEv1 (CVL)	A3343	Authentication Method - Digital Signature, Pre-shared Key Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 256, 384, 2048 Hash Algorithm - SHA2-256, SHA2-384 Preshared Key Length - Preshared Key Length: 8-256 Increment 8	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A3343	Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 256, 384, 2048 Derived Keying Material Length - Derived Keying Material Length: 1136-2432 Increment 8 Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
KDF SSH (CVL)	A3341	Cipher - AES-128, AES-192, AES-256, TDES Hash Algorithm - SHA-1, SHA2-256, SHA2-384	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A3342	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A3342	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A3342	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
Safe Primes Key Generation	A3342	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
Safe Primes Key Verification	A3342	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
SHA-1	A3342	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3335	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-256	A3339	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A3342	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3343	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-512	A3335	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A3340	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A3342	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG - Section 4	Key Type :Symmetric and Asymmetric	N/A	NIST SP800-133r2 Section 4: Symmetric key generation and Asymmetric seed generation using an unmodified output from an Approved DRBG (example 1); The module supports the following per NIST SP 800-133r2: 1. Section 5.1: Key Pairs for Digital Signature Schemes 2. Section 5.2: Key Pairs for Key Establishment 3. Section 6.2.1: Derivation of symmetric keys

Table 6: Vendor-Affirmed Algorithms

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 22.2R2-S2.3 - 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 22.2R2-S2.3 - 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

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	CKG KAS-135KDF KAS-Full KAS-SSC	Key Agreement for SSHv2	IG: IG D.F Scenario 2, path (2), split Key	KAS-ECC-SSC Sp800-56Ar3: (A3342) KDF SSH:

Name	Type	Description	Properties	Algorithms
			confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	(A3341) CKG - Section 4: () Key Type : Symmetric and Asymmetric
KAS2	CKG KAS-135KDF KAS-SSC	Key Agreement for SSHv2	IG: IG D.F Scenario 2, path (2), split Key confirmation:no Key derivation: IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides 112 bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A3342) KDF SSH: (A3341) Safe Primes Key Generation: (A3342) Safe Primes Key Verification: (A3342) CKG - Section 4: () Key Type : Symmetric and Asymmetric
KTS1	KTS-Wrap	Key Transport for SSHv2	Standard:SP 800-38F IG D.G: approved method from IG D.G Key confirmation:no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A3342) AES-CTR: (A3342) HMAC-SHA-1: (A3342) HMAC-SHA2-256: (A3342) HMAC-SHA2-512: (A3342) SHA-1: (A3342) SHA2-256: (A3342) SHA2-512: (A3342)
ECDSA SigVer	DigSig-SigVer	ECDSA Signature Verification used for identity-based public key authentication	FIPS 186-5:size: P-256, P-384, P-521 curves, 128, 192 and 256 bits	ECDSA SigVer (FIPS186-5): (A3342)

Name	Type	Description	Properties	Algorithms
DRBG	DRBG	Kernel DRBG providing random bits for SSP generation in the user/application space		HMAC DRBG: (A3335) HMAC-SHA2-256: (A3335) SHA2-256: (A3335)
Entropy Souce	ENT-Cond	Non-Physical Entropy Source		SHA2-512: (A3335)
ECDSA KeyGen	AsymKeyPair-KeyGen CKG	Generation of SSH host keys		ECDSA KeyGen (FIPS186-5): (A3342) CKG - Section 4: () Key Type : Symmetric and Asymmetric
ECDSA KeyGen2	AsymKeyPair-KeyGen CKG	SSP Agreement in the context of SSH		ECDSA KeyGen (FIPS186-5): (A3342) CKG - Section 4: () Key Type : Symmetric and Asymmetric
ECDSA KeyVer	AsymKeyPair-KeyVer	Verification of keys generated		ECDSA KeyVer (FIPS186-5): (A3342)
ECDSA SigGen	DigSig-SigGen	Signature Generation using ECDSA in the context of SSH		ECDSA SigGen (FIPS186-5): (A3342)
RSA KeyGen	AsymKeyPair-KeyGen CKG	Generation of SSH host keys		RSA KeyGen (FIPS186-5): (A3342) CKG - Section 4: () Key Type : Symmetric and Asymmetric
RSA SigGen	DigSig-SigGen	Signature Generation using RSA in the context of SSH		RSA SigGen (FIPS186-5): (A3342)
RSA SigVer	DigSig-SigVer	Signature Verification using RSA for		RSA SigVer (FIPS186-5): (A3342)

Name	Type	Description	Properties	Algorithms
		public key authentication		
Password Hash	SHA	Used to store passwords in hashed form		SHA2-512: (A3340)
KTS2	KTS-Wrap	Key Transport for IPsec	Standard:SP 800-38F IG D.G :approved method from IG D.G Key confirmation: no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A3343, A3339) HMAC-SHA2-256: (A3343, A3339) SHA2-256: (A3343, A3339)
KAS3	CKG KAS-135KDF KAS-Full KAS-SSC	Key Agreement in the context of IPsec	IG :IG D.F Scenario 2, path (2), split Key confirmation :no Key derivation :IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides 112 bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A3342) KDF IKEv1: (A3343) KDF IKEv2: (A3343) CKG - Section 4: () Key Type : Symmetric and Asymmetric Safe Primes Key Generation: (A3342) Safe Primes Key Verification: (A3342)
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: (A3342, A3343, A3339) HMAC-SHA-1: (A3342) HMAC-SHA2-256: (A3342, A3335, A3343, A3339) HMAC-SHA2-512: (A3342)

Name	Type	Description	Properties	Algorithms
				KAS-ECC-SSC Sp800-56Ar3: (A3342) KAS-FFC-SSC Sp800-56Ar3: (A3342) KDF SSH: (A3341) ECDSA SigGen (FIPS186-5): (A3342) ECDSA SigVer (FIPS186-5): (A3342) RSA SigGen (FIPS186-5): (A3342) RSA SigVer (FIPS186-5): (A3342) HMAC DRBG: (A3335) SHA2-512: (A3335) KDF IKEv1: (A3343) KDF IKEv2: (A3343)
KAS4	CKG KAS-135KDF KAS-Full KAS-SSC	Key Agreement in the context of IPsec	IG: IG D.F Scenario 2, path (2), split Key confirmation:no Key derivation: IG 2.4.B SP 800-135rev1 CVL Caveat :Key establishment methodology provides between 128 and 256 bits of security strength	KAS-ECC-SSC Sp800-56Ar3: (A3342) KDF IKEv1: (A3343) KDF IKEv2: (A3343) CKG - Section 4: ()

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
E56	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	Junos OS 22.2R2 on VMWare ESXi v7.0 with Intel(R) Xeon(R) CPU E5-2660 v4 (Broadwell) on HP ProLiant DL380 Gen9 Server; Junos OS 22.2R2 on VMWare ESXi v7.0 with Intel(R) Xeon(R) E-2254ML (Coffee Lake) on PacStar 451 Server	8 bits	6.4 bits	SHA2-512 (CAVP Cert. #A3335)

Table 11: Entropy Sources

2.9 Key Generation

The module implements an approved NIST SP 800-90Ar1 DRBG 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/IKEv1/IKEv2 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, KAS2, KAS3 and KAS4 in the Security Functions Implementations Table 9 have been documented in accordance with this requirement:

KAS1: KAS (KAS-ECC-SSC Cert. #A3342 and CVL Cert. #A3341; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

KAS2: KAS (KAS-FFC-SSC Cert.#A3342 and CVL Cert. #A3341; SSP establishment methodology provides 112 bits of encryption strength)

KAS3: KAS (KAS-FFC-SSC Cert.#A3342 and CVL Cert. #A3343; SSP establishment methodology provides 112 bits of encryption strength)

KAS4: KAS (KAS-ECC-SSC Cert. #A3342 and CVL Cert. #A3343; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

The Approved Algorithm list includes the tested components (KAS-ECC-SSC, KAS-FFC-SSC, KDF SSH, KDF IKEv1 and KDF IKEv2) as individual entries.

Per IG D.G:
The module supports the IETF SSH and IPsec protocols and thus implements key transport in the context of the protocols (per the KTS1 and KTS2 entries in the Security Functions Implementations Table 9).

The module implements the approved KTS using approved AES modes:

o KTS1: KTS (AES Cert. #A3342 and HMAC Cert. #A3342; key establishment methodology provides between 128 and 256 bits of encryption strength corresponding to the key lengths between 128 to 256 bits), used in the context of the IETF SSH protocol.

o KTS2: KTS (AES Certs. #A3339, #A3343 and HMAC Certs. #A3339, #A3343; SSP establishment methodology provides between 128 and 256 bits of encryption strength), used in the context of the IETF IKEv1/IKEv2 protocol

2.11 Industry Protocols

No parts of the SSH and IPsec protocols, other than the KDF SSH and the KDF IKEv1/KDF IKEv2 for IPsec, have been tested by the CAVP or CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Virtual Ethernet Ports, Virtual Serial Ports
N/A	Data Output	Virtual Ethernet Ports, Virtual Serial Ports
N/A	Control Input	Virtual Ethernet Ports, Virtual Serial Ports
N/A	Status Output	Virtual Ethernet Ports, Virtual Serial Ports

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 = 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	SHA2-512 (A3340)	1/(96^10)	9/(96^10)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	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 (FIPS186-5) (A3342)	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 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	RSA SigVer (FIPS186-5) (A3342)	1/ (2^112)	15,000/(2^112)

Table 13: Authentication Methods

The module enforces the separation of roles using role-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 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 Entropy Souce ECDSA KeyGen ECDSA KeyGen2 RSA KeyGen Password Hash	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 Authentication 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 -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 successful completion of each service	Commands (miscellaneous commands e.g., for IP address configuration, routing protocols, etc.)	Traffic	Password Hash	Super-user - CO Password: E Root - CO Password: E
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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
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 zeroisation)	Zeroise: Destroy all SSPs	successful deletion of virtual machine	Power (deletion of virtual machine)	N/A	Password Hash	Super-user - 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 - HMAC_DRBG entropy input: Z - HMAC_DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 - JuniperRootCA: 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 - IKE-PSK: Z - IKE-SKEYID: Z - IKE-SEK: Z - IKE-DH-PRI: Z - ESP-SEK: Z - IKE-DH-PUB: Z Root - SSH Private Host Key: Z - SSH ECDH Private Key: Z - SSH DH Private Key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 - JuniperRootCA: Z - PackageCA: Z - SSH ECDH Public Key: Z - SSH DH Public Key: Z - SSH ECDH Client Public Key: Z - SSH DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Client Public Key: Z - IKE-PSK: Z - IKE-SKEYID: Z - IKE-SEK: Z - ESP-SEK: Z - IKE-DH-PRI: Z - IKE-DH-PUB: Z
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 SigVer DRBG Entropy Souce ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen RSA SigVer Password Hash	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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH ECDH 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 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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH ECDH Public Key: G,E,Z - SSH DH 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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 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 Shared Secret: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 each service	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 Root - CO Password: E
Perform self-tests (remote reset)	Software initiated reset, performs self-tests on	Global Approved Mode indicator "fips" at the CLI	Control input/reset signal (request vmhost reboot)	N/A	KAS1 KAS2 KTS1 DRBG Entropy Souce	Super-user - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	demand via SSH	combined with successful completion of each service			ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen Password Hash CASTs on boot	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 - Software Integrity Key: E - SSH Private Host Key: E - SSH Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E Root - SSH ECDH Private Key: Z - SSH DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 - Software Integrity Key: E - SSH Private Host Key: E - SSH Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
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 - Software Integrity Key: E Root - Software Integrity Key: E Operator - Software Integrity Key: E Read-only - Software Integrity Key: E Unauthorised - Software Integrity Key: E Unauthenticated - Software Integrity Key: E
Perform approved security functions (IPsec connection)	Initiate IPsec connection	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Commands (set security ipsec security-association sa-name; * set interfaces <name> unit 0 family inet address <ip address>; * set security ike security-association sa-name)	IPsec session	KTS2 KAS3 KAS4	Root - IKE-PSK: W,E - IKE-SKEYID: G,E,Z - IKE-SEK: G,E,Z - ESP-SEK: G,E,Z - IKE-DH-PRI: G,E,Z - IKE-DH-PUB: G,R,E,Z Super-user - IKE-PSK: W,E - IKE-SKEYID: G,E,Z - IKE-SEK: G,E,Z - ESP-SEK: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IKE-DH-PRI: G,E,Z - IKE-DH-PUB: G,R,E,Z

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 EC Diffie-Hellman with ed25519 curve ARCFOUR Blowfish CAST DSA (SignGen, SigVer, non-compliant) HMAC-MD5 HMAC-RIPEMD160 UMAC	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

Table 16: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support software loading from an external source.

4.6 Cryptographic Output Actions and Status

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

- set security ipsec security-association *sa-name*
- set security ike security-association *sa-name*

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

- show security ike security-associations
- show security ipsec security-associations
- show security ipsec statistics

5 Software/Firmware Security

5.1 Integrity Techniques

The module performs the software integrity check using ECDSA P-256 with SHA2-256 (CAVP Cert. #A3342). 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) until zeroisation of the module.

5.2 Initiate on Demand

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

5.3 Additional Information

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

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The module contains a modifiable operational environment since the underlying hardware platform supports uncontrollable modifications to itself. The module contains the operating system Junos OS 22.2R2-S2.3.

6.2 Configuration Settings and Restrictions

Security rules and restrictions for configuration of the operational environment have been specified in Sections 11.1 and 11.4 of this document.

7 Physical Security

The requirements per this section do not apply since the module is of type software.

8 Non-Invasive Security

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 17: 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	External endpoint	NVRAM	Plaintext	Manual	Direct	

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
connection - NVRAM						
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	
Output during IPsec negotiation	RAM	External endpoint	Plaintext	Automated	Electronic	

Table 18: 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/IPsec respectively.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Deletion of virtual instance	Deletion of the vSRX 3.0 instance	Used to provide zeroisation as a service	Operator initiated
Power-cycle	Power cycling the underlying host platform to zeroise temporary SSPs	Power cycling the underlying host platform to zeroise temporary SSPs	Operator initiated
Session termination	Termination of sessions automatically zeroises temporary SSPs used as part of the session	Termination of sessions automatically zeroises temporary SSPs used as part of the session	Module initiated
Derivation of session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of session key	Module initiated

Table 19: 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	DRBG 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-384, P-512 - 128 bits, 192 bits, 256 bits	ECDH Private Key - CSP	DRBG ECDSA KeyGen2		KAS1
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	DRBG		KAS2
SSH Session Key	SSH Session Key	128 bits, 192 bits, 256 bits - 128 bits, 192 bits, 256 bits	Session Key - CSP		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			

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 bits - 256 bits	Internal state of the DRBG - CSP	DRBG		DRBG
HMAC_DRBG Key value	A critical value of the internal state of DRBG	440 bits - 440 bits	Internal state of the DRBG - CSP	DRBG		DRBG
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		DRBG
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	DRBG ECDSA KeyGen RSA KeyGen		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
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 RSA - 128, 192, 256 bits for ECDSA, 112, 192 and 256 bits for RSA	Public key - PSP			
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			
JuniperRootCA	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	ECDSA P-256 - 128 bits	Public key certificate - Neither			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	public key for the signing key used to generate all the signatures used on the packages and signature lists					
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	DRBG 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	DRBG		
Software Integrity Key	Public key used to perform the software integrity test on each boot	ECDSA P-256 - 128 bits	Public key - Neither			
IKE-PSK	Pre-Shared Key used to authenticate IKE connections	256 bits - 256 bits	IKE Pre-Shared Key - CSP			
IKE-SKEYID	IKE secret used to derive IKE and IPsec ESP session keys	256 bits - 256 bits	IKE shared secret - CSP		KAS3 KAS4	KAS3 KAS4
IKE-SEK	IKE Session Keys. AES	AES: 128 bits,	IKE Session Key - CSP		KAS3 KAS4	KTS2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	(128 bits), HMAC (SHA-256)	HMAC: 256 bits - AES: 128 bits, HMAC: 256 bits				
ESP-SEK	ESP Session Keys. AES (128 bits), HMAC (SHA-256)	AES: 128 bits, HMAC: 256 bits - AES: 128 bits, HMAC: 256 bits	ESP Session Key - CSP		KAS3 KAS4	KTS2
IKE-DH-PRI	Diffie-Hellman private key used in IKE	2048 bits - 112 bits	IKE Diffie-Hellman private key - CSP	KAS3 KAS4		
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			
IKE-DH-PUB	Diffie-Hellman public key used in IKE	2048 bits - 112 bits	IKE Diffie-Hellman public key - PSP	KAS3 KAS4		

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH Private Host Key		NVRAM:Plaintext		Deletion of virtual instance	
SSH ECDH Private Key		RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
SSH DH Private Key		RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
SSH Session Key		RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
User Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated		Deletion of virtual instance	
CO Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated		Deletion of virtual instance	
HMAC_DRBG V value		RAM:Plaintext	Until power-cycle	Power-cycle	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
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	Deletion of virtual instance Power-cycle Derivation of session key	
DH Shared Secret		RAM:Plaintext	Until SSH session key derivation	Deletion of virtual instance Power-cycle Derivation of session key	
HMAC Key		RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
SSH Public Host Key	Output during SSH negotiation (host key)	NVRAM:Plaintext		Deletion of virtual instance	
User Authentication Public Keys	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Deletion of virtual instance	
CO Authentication Public Keys	Entered over SSH - NVRAM	NVRAM:Plaintext		Deletion of virtual instance	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	Entered through the CLI via console connection - NVRAM				
JuniperRootCA	Loaded at manufacture	NVRAM:Plaintext		Deletion of virtual instance	
PackageCA	Loaded at manufacture	NVRAM:Plaintext		Deletion of virtual instance	
SSH ECDH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
SSH DH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Deletion of virtual instance Power-cycle Session termination	
Software Integrity Key	Loaded at manufacture	NVRAM:Plaintext		Deletion of virtual instance	
IKE-PSK	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Deletion of virtual instance	
IKE-SKEYID		RAM:Plaintext	until session key derivation	Derivation of session key	
IKE-SEK		RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Session termination	
ESP-SEK		RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle Session termination	
IKE-DH-PRI		RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle Session termination	IKE-DH-PUB:Paired With
SSH ECDH Client Public Key	Input during SSH negotiation	RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle Session termination	
SSH DH Client Public Key	Input during SSH negotiation	RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle Session termination	
IKE-DH-PUB	Output during IPsec negotiation	RAM:Plaintext	until session termination	Deletion of virtual instance Power-cycle Session termination	IKE-DH-PRI:Paired With

Table 21: SSP Table 2

9.5 Transitions

Per the NIST SP 800-133Ar2/3 and the programmatic transitions defined by the CMVP, the following algorithm transitions apply to the module, and the algorithms have been designated allowed/non-approved accordingly in Section 2.5:

a. Usage of SHA-1 for SigVer is allowed for legacy use only until 2030. Thereafter, all usage of SHA-1 will be considered a non-approved, not allowed algorithm.
b. Until January 1, 2031, the following algorithms will be considered deprecated:
    a. Hash function and HMAC using SHA-1 hash function
    b. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation
c. As of January 1, 2031, the following algorithms will be considered deprecated/disallowed (i.e. non-approved, not allowed)/legacy use:
    a. Use of the 112-bit security strength for classical digital signature and key-establishment mechanisms (deprecated)
    b. Use of the 112-bit security strength for block ciphers (disallowed)
    c. Use of a security strength less than 128-bits but greater than 112 bits for ECDA KeyGen and RSA KeyGen (PKCS #1 v1.5 & PSS) (deprecated)
    d. HMAC using SHA-1 hash function (legacy use)
    e. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation (disallowed)
    f. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Verification (legacy use)

10 Self-Tests

10.1 Pre-Operational Self-Tests

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

Table 22: 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 software integrity test on each boot prior to executing the software integrity test.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG (A3335)	Prediction Resistance: Yes Supports Reseed Capabilities: Mode: SHA2-256 Entropy Input: 256 Nonce: 128 Personalizati	KAT	CAST	NIST 800-90 HMAC DRBG Known Answer Test : Passed	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	on String Length: 0-256 Increment 8 Additional Input: 8-256 Increment 8 Returned Bits: 1024					
HMAC-SHA2-256 (A3335)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
AES-CBC (A3342) - Encrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A3342) - Encrypt - 192 bits	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A3342) - Encrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A3342) - Decrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
AES-CBC (A3342) - Decrypt - 192 bits	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A3342) - Decrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
HMAC-SHA-1 (A3342)	Key Length: 160 bits	KAT	CAST	HMAC-SHA-1 Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-256 (A3342)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-512 (A3342)	Key Length: 512 bits	KAT	CAST	HMAC-SHA2-512 Known Answer Test : Passed	N/A	During boot
KAS-ECC-SSC Sp800-56Ar3 (A3342) - P-256	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 (A3342) - P-384	Domain Parameter Generation Methods: P-384	KAT	CAST	KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed	N/A	During boot
KAS-FFC-SSC	Domain Parameter Generation	KAT	CAST	KAS-FFC-EPHEM-	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Sp800-56Ar3 (A3342)	Methods: MODP-2048			NOKC Known Answer Test: Passed		
KDF SSH (A3341)	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 (FIPS186-5) (A3342)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-SIGN Known Answer Test: Passed	Sign	During boot
RSA SigVer (FIPS186-5) (A3342)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-VERIFY Known Answer Test: Passed	Verify	During boot
ECDSA SigGen (FIPS186-5) (A3342)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	CAST	ECDSA-SIGN Known Answer Test: Passed	Sign	During boot
ECDSA SigVer (FIPS186-5) (A3342)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	CAST	ECDSA-VERIFY Known Answer Test: Passed	Verify	During boot
SHA2-512 (A3340)	SHA2-512	KAT	CAST	SHA-2-512 Known Answer Test: Passed	N/A	During boot
Entropy test - NIST SP	NIST SP 800-90B Repetitive Count Test	RCT	CAST	pass	Cutoff value C = 21	During boot and continually

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
800-90B RCT						
Entropy test - NIST SP 800-90B APT	NIST SP 800-90B Adapative Proportion Test	APT	CAST	pass	W = 512; Cutoff value C = 311	During boot and continually
ECDSA KeyGen (FIPS186-5) (A3342)	Curve: P-256 Hash Algorithm: SHA2-256	PCT	PCT	0	Key pair generated for SSP agreement in the context of SSHv2 protocol and for key generation for use in ECDSA signature generation/verification	On key generation
KAS-FFC-SSC Sp800-56Ar3 (A3342) - PCT	Capabilities: Domain Parameter: MODP2048	PCT	PCT	0	Key pair generated for SSP agreement in the context of SSHv2 protocol	On key generation
RSA KeyGen (FIPS186-5) (A3342)	Modulus: 2048 Hash SHA2-256	PCT	PCT	0	Key pair generated for signature generation/verification in the context of SSHv2 protocol	On key generation
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and IKE-PSK via direct connection to the module's console (serial) interface	Duplicate entry test required for entry of operator passwords and IKE-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 IKE-PSK

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF IKEv1 (A3343)	IKEv1 (IPSec) KDF	KAT	CAST	IKEV1 Known Answer Test: Passed	N/A	During boot
KDF IKEv2 (A3343)	IKEv2 (IPSec) KDF	KAT	CAST	IKEV2 Known Answer Test: Passed	N/A	During boot
AES-CBC (A3343) - Encrypt - 128 bits	Key length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A3343) - Decrypt - 128 bits	Key length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decryt	During boot
HMAC-SHA2-256 (A3343)	Key length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
AES-CBC (A3339) - Encrypt - 128 bits	Key length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A3339) - Decrypt - 128 bits	Key length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
HMAC-SHA2-256 (A3339)	Key length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Test : Passed		

Table 23: 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 DSA tests respectively). The software load test is performed when a software image (.tgz) 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
Software Integrity Test	KAT	SW/FW Integrity	On Demand	Manually via a reboot

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A3335)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A3335)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3342) - Encrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3342) - Encrypt - 192 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3342) - Encrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3342) - Decrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3342) - Decrypt - 192 bits	KAT	CAST	On Demand	Manually via a reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A3342) - Decrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA-1 (A3342)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A3342)	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-512 (A3342)	KAT	CAST	On Demand	Manually via a reboot
KAS-ECC-SSC Sp800-56Ar3 (A3342) - P-256	KAT	CAST	On Demand	Manually via a reboot
KAS-ECC-SSC Sp800-56Ar3 (A3342) - P-384	KAT	CAST	On Demand	Manually via a reboot
KAS-FFC-SSC Sp800-56Ar3 (A3342)	KAT	CAST	On Demand	Manually via a reboot
KDF SSH (A3341)	KAT	CAST	On Demand	Manually via a reboot
RSA SigGen (FIPS186-5) (A3342)	KAT	CAST	On Demand	Manually via a reboot
RSA SigVer (FIPS186-5) (A3342)	KAT	CAST	On Demand	Manually via a reboot
ECDSA SigGen (FIPS186-5) (A3342)	KAT	CAST	On Demand	Manually via a reboot
ECDSA SigVer (FIPS186-5) (A3342)	KAT	CAST	On Demand	Manually via a reboot
SHA2-512 (A3340)	KAT	CAST	On Demand	Manually via a reboot
Entropy test - NIST SP 800-90B RCT	RCT	CAST	On Demand	Manually via a reboot
Entropy test - NIST SP 800-90B APT	APT	CAST	On Demand	Manually via a reboot
ECDSA KeyGen (FIPS186-5) (A3342)	PCT	PCT	On Demand	Manually via a reboot
KAS-FFC-SSC Sp800-56Ar3 (A3342) - PCT	PCT	PCT	On Demand	Manually via a reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA KeyGen (FIPS186-5) (A3342)	PCT	PCT	On Demand	Manually via a reboot
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and IKE-PSK via direct connection to the module's console (serial) interface	Manual Entry	On Demand	Manually via configuration of operator passwords and IKE-PSK
KDF IKEv1 (A3343)	KAT	CAST	On Demand	Manually via a reboot
KDF IKEv2 (A3343)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3343) - Encrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3343) - Decrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A3343)	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3339) - Encrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
AES-CBC (A3339) - Decrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot
HMAC-SHA2-256 (A3339)	KAT	CAST	On Demand	Manually via a reboot

Table 25: Conditional Periodic Information

The pre-operational software 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 software 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 software integrity test or if any of the CASTs fail	N/A	"FIPS error: self-test failure" for software 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 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 APT or RCT test fails	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	Entropy data discarded in case of APT/RCT failure

Table 26: Error States

If the pre-operation software 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 module is in the non-compliant state by default. The Crypto Officer (CO) shall follow the instructions in this section to download, install and initialize the module onto the host platforms identified in Table 2. Next, the module must be configured in Approved mode, as described below, and rebooted. Once the module is rebooted and the integrity and self-tests have run successfully on initial boot, the module is operational in the Approved mode.

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

Downloading the Image

The Crypto Officer must check to verify the 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.

Guide to Download Software Packages for vSRX 3.0 from Juniper Networks:
1. Using a Web browser, follow the link to the download URL on the Juniper Networks webpage at https://www.juniper.net/support/downloads/?p=vsrx#sw
2. Log in to the Juniper Networks website using the username (generally your e-mail address) and password supplied by your Juniper Networks representatives.
3. Under "Version" dropped down list, select the appropriate certified Release (Example: 22.2R2-S2.3).
4. Under "Application Media" section, select the appropriate software package for the target release version and hypervisor.
5. Download Junos OS to a local host or to an internal software distribution site.
6. MD5 checksum and SHA1 checksum can be found under "Checksum"
o Verify the checksum of the download with the provided checksum
The crypto-officer shall follow the instructions for installation provided in the Juniper Networks documentation: For installing the vSRX 3.0 using a .tgz file, the instructions can be found in the Junos® OS FIPS Evaluated Configuration Guide for vSRX 3.0 Instance. The CLI command from the aforementioned document to install Junos OS is repeated below:

```
>request system software add /<image-path>/<junos package>no-copy no-validate reboot
```

Where the <junos package> is the .tgz file for e.g. junos-install-vsrx3-x86-64-22.2R2.tgz.
For installing the vSRX 3.0 using an .ova file, the instructions can be found in the vSRX Guide for VMware.

The steps from the aforementioned document are repeated below:
1. Enter the vCenter server hostname or address in your browser (https://<ipaddress>:9443) to access the vSphere WebClient, and login to the vCenter server with your credentials.
2. Select a host or other valid parent for a virtual machine and click Actions>All vCenter Actions>Deploy OVF Template.
3. Click Browse to locate the vSRX 3.0 software package, and then click Next.
4. Click Next in the OVF Template Details window.
5. Click Accept in the End User License Agreement window, and then click Next.
6. Change the default vSRX 3.0 VM name in the Name box and click Next. It is advisable to keep this name the same as the hostname you intend to give to the VM.
7. In the Datastore window, do not change the default settings for:
- Datastore
- Available Space

8. Select a datastore to store the configuration file and virtual disk files in OVF template, and then click Next.
9. Select your management network from the list, and then click Next. The management network is assigned to the first network adapter, which is reserved for the management interface (fxp0).
10. Click Finish to complete the installation.
11. Open the Edit Settings page of the vSRX 3.0 VM and select a virtual switch for each network adapter. Three network adapters are created by default. Network adapter 1 is for the management network (fxp0). To add a fourth adapter, select Network from New device list at the bottom of the page.
12. Enable promiscuous mode for the management virtual switch:
1. Select the host where the vSRX 3.0 VM is installed and select Manage>Networking >Virtual switches.
2. In the list of virtual switches, select vSwitch0 to view the topology diagram for the management network connected to network adapter 1.
3. Click the Edit icon at the top of the list, select Security, and select Accept next to Promiscuous mode. Click OK.

Once the FIPS 140-3 validated vSRX 3.0 software is installed on the hardware platform and hypervisor in Table 3 then the crypto-officer shall follow the instructions below to place the module in the Approved mode of operation.

Enabling the Approved Mode of Operation:

The CO shall enable the module for Approved mode of operation by performing the following steps.

1. Set up the password for root authentication:

```
root@host> set system root-authentication plaintext-password
Enter password:
Re-enter password:

root@host> commit
commit complete
```

2. Enable the Approved mode:

```
root@host> set system fips level 2 *
```

3. Commit.

```
root@host> commit
```

4. Restart the virtual machine from the hypervisor console.
5. Verify that the module is operational in the Approved mode in the correct version (Junos OS 22.2R2-S2.3) by using the “show version” command. The command prompt should contain the string “fips” denoting that the Approved mode has been successfully configured.

*Note: This module is a FIPS 140-3 Security Level 1 module but the command “set system fips level 2” must be used to invoke the Approved mode of operation. Please note this is Juniper terminology only. The module claims to meet FIPS 140-3/ISO 19790 Security Level 1.

Then, the CO must run the following commands to configure the SSHv2 protocol:

1. Edit system services ssh and set root-login allow

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

2. Assign an IP address to the fxp0 interface and set the routing options

```
[edit]
root@host# set interfaces fxp0 unit <unit> family inet address <ip address>
root@host# set routing-options static route 0.0.0.0/0 next-hop <gateway>
```

The “show configuration security ike” and “show configuration security ipsec” commands display the approved and configured IKE/IPsec configuration for the module.

Zeroisation
The vSRX 3.0 instance can be deleted from the datastore of the VMware ESXi 7.0 hypervisor as the method of zeroisation. The cryptographic officer must retain control of the module while zeroisation is in process.

11.2 Administrator Guidance

For further information and for the Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for vSRX, Release 22.2R2 document.

11.3 Non-Administrator Guidance

For further information and for the non-Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for vSRX, Release 22.2R2 document.

11.4 Design and Rules

The module design corresponds to the security rules below. The term must 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. Pre-operational self-test do not require any operator action.
4. Data output is inhibited during key generation, self-tests, zeroization, and error states.
5. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
6. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.

7. The module does not support a maintenance interface or role.
8. The module does not support manual key entry.
9. The module does not output intermediate key values.
10. The module does not output plaintext CSPs.
11. The cryptographic officer must retain control of the module while zeroisation is in process.
12. Entropy Source: The Juniper Networks vSRX 3.0 Virtual Firewall (running Junos OS 22.2R2-S2.3) cryptographic module is a software module with an entropy gathering non-physical entropy source inside of the module's physical perimeter per IG 9.3.A Scenario 1. (b) (ESV cert.#E56 applies to the module). The module generates sufficient entropy for the generation of SSPs (using the approved DRBG of the module) with the maximum target security strength (256 bits) needed. As can be verified from the Public Use Document and Security Policy Section 2.8, Entropy Sources Table 11, the entropy source generates a minimum of 448 bits of overall entropy per 512-bit output sample/ 6.4 per 8-bit sample (entropy input to the DRBG). The DRBG is seeded with 512 bits.

11.5 Maintenance Requirements

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

11.6 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.