Juniper Networks vSRX Virtual Firewall

Non-Proprietary FIPS 140-2 Cryptographic Module Security Policy

Version: 1.3

Date: February 13, 2018

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

Table of Contents

List of Tables

List of Figures

1 Introduction

The Juniper Networks vSRX Virtual Firewall (here after referred to as vSRX or the module) is a secure firewall that provide 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. The vSRX runs Juniper's JUNOS software. The JUNOS software is FIPS-compliant, when configured in FIPS-MODE called JUNOS-FIPS-MODE, version 15.1X49-D100. The software image is junos-srxmr-15.1X49-D100.6-domestic.tgz for the vSRX and the software status service identifies itself as in the "Junos 15.1X49-D100.6".

The cryptographic module is defined as multiple-chip standalone software module. The module executes JUNOS-FIPS software on an VMware ESXi Hypervisor on the Server HP ProLiant DL380 Gen9 physical platform.

Table 1 – Cryptographic Module Tested Configurations

Model	Software Version	Processor	HypervisorESXi	Hardware Platform
vSRX	JUNOS 15.1X49-D100	Intel(R) Xeon(R) E5	ESXi 5.5	Server HP ProLiant DL380 Gen9

The module is designed to meet FIPS 140-2 Level 1 overall:

Table 2 – Security Level of Security Requirements

Area	Description	Level
1	Module Specification	1
2	Ports and Interfaces	1
3	Roles and Services	3
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Key Management	1
8	EMI/EMC	1
9	Self-test	1
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A
	Overall	1

The module has a limited operational environment as per the FIPS 140-2 definitions. The module does not implement any mitigations of other attacks as defined by FIPS 140-2.

1.1 Cryptographic Boundary

The cryptographic boundary of the module is depicted in Figure 1 below. The physical cryptographic boundary is defined as the outer edge of the hardware server on which the hypervisor and Juniper Networks vSRX Virtual Firewall are installed. The module does not rely on external devices for input and output. The logical boundary is the Juniper vSRX Virtual Firewall which is comprised of the Junos 15.1X49 D-100 software.

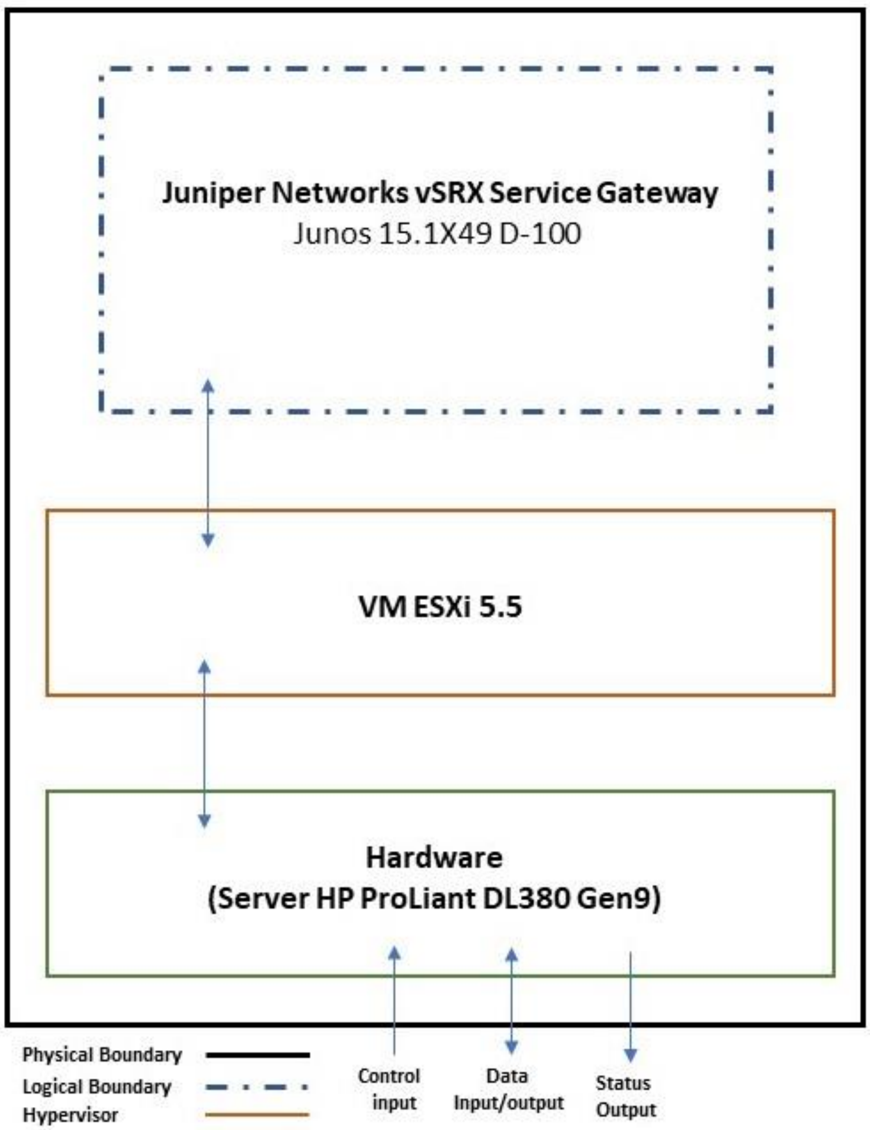

Figure 1- Module's Cryptographic Boundary

Table 3 – Ports and Interfaces

Physical Port/Interface	Logical Port/Interface	FIPS Interface
Host Platform Ethernet ports	Virtual Ethernet Ports	Data Input
Host Platform Ethernet ports	Virtual Ethernet Ports	Data Output
Host Platform Ethernet ports/ Serial port	Virtual Ethernet Ports, Virtual Serial Ports	Control Input
Host Platform Ethernet ports/ Serial port	Virtual Ethernet Ports, Virtual Serial Ports	Status Output

1.2 Mode of Operation

The Crypto-Officer (CO) shall follow the instructions in Section 6 to download, install and initialize the module onto the platform identified in Table 1. Next, the module is configured in FIPS-MODE and rebooted. Once the module is rebooted and the integrity and self-tests have run successfully on initial power-on in FIPS-MODE, the module is operating in the FIPS-Approved mode.

If the module was previously in a non-Approved mode of operation, the Cryptographic Officer must zeroize the CSPs by following the instructions in Section 1.3

The CO shall enable the module in FIPS mode or operation by performing the following steps.

1. Enable the FIPS mode on the device.
   user@host> set system fips level 2
2. Commit and reboot the device.
   user@host> commit

Note: This module is a FIPS Level 1 module but the command "set system fips level 2" must be used to invoke a FIPS mode of operation.

Then, the CO must run the following commands to configure SSH to use FIPS approved and FIPS allowed algorithms:

1. Specify the permissible SSH host-key algorithms for the system services.
   [edit system services]
   root@host# set ssh hostkey-algorithm ssh-ecdsa

2. Specify the SSH key-exchange for Diffie-Hellman keys for the system services.
   [edit system services]
   root@host#set ssh key-exchange ecdh-sha2-nistp256

3. Specify all the permissible message authentication code algorithms for SSHv2.
   [edit system services]
   root@host#set ssh macs hmac-sha1

4. Specify the ciphers allowed for protocol version 2.
   [edit system services]

root@host#set ssh ciphers aes128-cbc

When AES GCM is configured as the encryption-algorithm for IKE or IPsec, the CO must configure the module to use IKEv2 by running the following commands:

IKE:

root@host# set security ike proposal <ike_proposal_name> encryption-algorithm aes-256-gcm

IPSec:

root@host# set security ipsec proposal <ipsec_proposal_name> encryption-algorithm aes-128-gcm

root@host# set security ike gateway <gateway_name> version v2-only

root@host# commit
commit complete

When Triple-DES is configured as the encryption-algorithm for IKE or IPsec, the CO must configure the IPsec proposal lifetime-kilobytes to comply with [IG A.13] using the following command:

co@fips-srx:fips# set security ipsec proposal <ipsec_proposal_name> lifetime-kilobytes <kilobytes>"

co@fips-srx:fips# commit

When Triple-DES is the encryption-algorithm for IKE (regardless of the IPsec encryption algorithm), the lifetime-kilobytes for the associated IPsec proposal must be greater than or equal to 12800.

When Triple-DES is the encryption-algorithm for IPsec, the lifetime-kilobytes must be less than or equal to 33554432.

The "show version" command will indicate if the module is operating in FIPS mode (e.g. JUNOS Software Release [15.1X49-D100]) along with "fips" prompt.

The "show configuration security ike" and "show configuration security ipsec" commands display the approved and configured IKE/IPsec configuration for the device operating in FIPS-approved mode.

1.3 Zeroization

The cryptographic module provides a non-Approved mode of operation in which non-approved cryptographic algorithms are supported. When transitioning between the non-Approved mode of operation and the Approved mode of operation, the Cryptographic Officer must run the following commands to zeroize the Approved mode CSPs:

user@host> request system zeroize hypervisor

This command wipes clean all the CSPs/configs as well as the disk. Currently the device will have to be reimaged to bring back the device, as all the disk partitions are securely erased.

Use of the zeroize command is restricted to the Cryptographic Officer. The cryptographic officer shall perform zeroization in the following situations:

1. Before FIPS Operation: To prepare the device for operation as a FIPS cryptographic module by erasing all CSPs and other user-created data on a device before its operation as a FIPS cryptographic module.
2. Before non-FIPS Operation: To conduct erasure of all CSPs and other user-created data on a device in preparation for repurposing the device for non-FIPS operation.

Note: The Cryptographic Officer must retain control of the module while zeroization is in process.

2 Cryptographic Functionality

The module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in Tables 4, 5, 6, 7, 8 and 9 below.

Allowed Protocols

Table 10 summarizes the high-level protocol algorithm support.

2.1 Approved Algorithms

References to standards are given in square bracket [ ]; see the References table.

Table 4 – Data Plane Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
4723	AES [197]	CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		GCM [38D]	Key Sizes: 128, 192, 256	Encrypt, Decrypt, AEAD
3140	HMAC [198]	SHA-1	Key size: 160 bits, λ = 96	Message Authentication
		SHA-256	Key size: 256 bits, λ = 128	
3868	SHS [180]	SHA-1 SHA-256		Message Digest Generation
2505	Triple-DES [67]	TCBC [38A]	Key Size: 192	Encrypt, Decrypt

Table 5 – Control Plane QuickSec Approved Cryptographic Functions

Cert	Algorithm	Mode	Description	Functions
4719	AES [197]	CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		GCM [38D]	Key Sizes: 128, 256	Encrypt, Decrypt, AEAD
N/A[1]	CKG	[133] Section 6.2		Asymmetric key generation using unmodified DRBG output
		[133] Section 7.3		Derivation of symmetric keys
1362	CVL	IKEv1 [135]	SHA 256, 384	Key Derivation
		IKEv2 [135]	SHA 256, 384	
1608	DRBG [90A]	HMAC	SHA-256	Random Bit Generation
1166	ECDSA [186]		P-256 (SHA 256) P-384 (SHA 384)	KeyGen, SigGen, SigVer
3136	HMAC [198]	SHA-256	Key size: 256 bits,	

[1] Vendor Affirmed.

		SHA-384	λ = 128, 256	Message Authentication, KDF Primitive
			Key size: 384 bits, λ = 192, 384	
N/A	KTS	AES Cert. #4719 and HMAC Cert. #3136		key establishment methodology provides between 128 and 256 bits of encryption strength
		Triple-DES Cert. #2501 and HMAC Cert. #3136		key establishment methodology provides 112 bits of encryption strength
2572	RSA [186]	PKCS1_V1_5	n=2048 (SHA 256) n=4096 (SHA 256)	SigGen, SigVer[2]
3864	SHS [180]	SHA-256 SHA-384		Message Digest Generation
2501	Triple-DES [67]	TCBC [38A]	Key Size: 192	Encrypt, Decrypt

Table 6 – OpenSSL Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
4720	AES [197]	CBC [38A] CTR [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
1609	DRBG [90A]	HMAC	SHA-256	Random Bit Generation
N/A[3]	CKG	[133] Section 6.1 [133] Section 6.2		Asymmetric key generation using unmodified DRBG output
1167	ECDSA [186]		P-256 (SHA 256) P-384 (SHA 384)	SigGen, KeyGen, SigVer
3137	HMAC [198]	SHA-1	Key size: 160 bits, λ = 160	Message Authentication
		SHA-384[4]	N/A	
		SHA-512	Key size: 512 bits, λ = 512	
		SHA-256	Key size: 256, λ = 256	Message Authentication DRBG Primitive
N/A	KTS	AES Cert. #4720 and HMAC Cert. #3137		key establishment methodology provides between 128 and 256 bits of encryption strength
		Triple-DES Cert. #2502 and HMAC Cert. #3137		key establishment methodology provides 112 bits of encryption strength

[2] RSA 4096 SigVer was not tested by the CAVP; however, it is Approved for use per CMVP guidance, because RSA 2048 SigVer was tested and testing for RSA 4096 SigVer is not available.

[3] Vendor Affirmed.

[4] HMAC-SHA384 was validated; however, it is not used by any service.

2573	RSA [186]		n=2048 (SHA 256) n=4096 (SHA 256)	KeyGen[5] , SigGen, SigVer[6]
3865	SHS [180]	SHA-1 SHA-256 SHA-384		Message Digest Generation, KDF Primitive
		SHA-512		Message Digest Generation
2502	Triple-DES [67]	TCBC [38A]	Key Size: 192	Encrypt, Decrypt

Table 7 – OpenSSH Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
N/A[7]	CKG	[133] Section 7.3		Derivation of symmetric keys
1391	CVL	SSH [135]	SHA 1, 256, 384	Key Derivation

Table 8 – LibMD Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
3895	SHS [180]	SHA-256 SHA-512		Message Digest Generation

2.2 Allowed Algorithms

Table 9 – Allowed Cryptographic Functions

Algorithm	Caveat	Use
Diffie-Hellman [IG] D.8	Provides 112 bits of encryption strength.	key agreement; key establishment
Elliptic Curve Diffie-Hellman [IG] D.8	Provides 128 or 192 bits of encryption strength.	key agreement; key establishment
NDRNG [IG] 7.14 Scenario 1b	The module generates a minimum of 256 bits of entropy for key generation.	Seeding the DRBG

[5] RSA 4096 KeyGen was not tested by the CAVP; however, it is Approved for use per CMVP guidance, because RSA 2048 KeyGen was tested and testing for RSA 4096 KeyGen is not available.

[6]RSA 4096 SigVer was not tested by the CAVP; however, it is Approved for use per CMVP guidance, because RSA 2048 SigVer was tested and testing for RSA 4096 SigVer is not available.

[7] Vendor Affirmed.

2.3 Allowed Protocols

Table 10 – Protocols Allowed in FIPS Mode

<table>
<tr><th>Protocol</th><th>Key Exchange</th><th>Auth</th><th>Cipher</th><th>Integrity</th></tr>
<tr><td>IKEv1</td><td>Diffie-Hellman (L = 2048, N = 224, 256)
EC Diffie-Hellman P-256, P-384</td><td>RSA 2048
RSA 4096
Pre-Shared Secret
ECDSA P-256
ECDSA P-384</td><td>Triple-DES CBC
AES CBC 128/192/256</td><td>HMAC-SHA-1-96
HMAC-SHA-256-128
HMAC-SHA-384-192</td></tr>
<tr><td>IKEv2[8]</td><td>Diffie-Hellman (L = 2048, N = 224, 256)
EC Diffie-Hellman P-256, P-384</td><td>RSA 2048
RSA 4096
Pre-Shared Secret
ECDSA P-256
ECDSA P-384</td><td>Triple-DES CBC
AES CBC 128/192/256
AES GCM[9] 128/256</td><td>HMAC-SHA-1-96
HMAC-SHA-256-128
HMAC-SHA-384-192</td></tr>
<tr><td rowspan="2">IPsec ESP</td><td>IKEv1 with optional:
• Diffie-Hellman (L = 2048, N = 224, 256)
• EC Diffie-Hellman P-256, P-384</td><td>IKEv1</td><td>3 Key Triple-DES CBC
AES CBC 128/192/256</td><td rowspan="2">HMAC-SHA-1-96
HMAC-SHA-256-128</td></tr>
<tr><td>IKEv2 with optional:
• Diffie-Hellman (L = 2048, N = 224), (2048, 256)
• EC Diffie-Hellman P-256, P-384</td><td>IKEv2</td><td>3 Key Triple-DES CBC
AES CBC 128/192/256
AES GCM[10] 128/192/256</td></tr>
<tr><td>SSHv2</td><td>Diffie-Hellman (L = 2048, N = 256)
EC Diffie-Hellman P-256, P-384</td><td>ECDSA P-256</td><td>Triple-DES CBC
AES CBC 128/192/256
AES CTR 128/192/256</td><td>HMAC-SHA-1-96
HMAC-SHA-1
HMAC-SHA-256
HMAC-SHA-512</td></tr>
</table>

No part of these protocols, other than the KDF, have been tested by the CAVP and CMVP.

The IKE and SSH algorithms allow independent selection of key exchange, authentication, cipher and integrity. In reference to the Allowed Protocols in Table 10 above: each column of options for a given protocol is independent, and may be used in any viable combination. These security functions are also available in the SSH connect (non-compliant) service.

[8] IKEv2 generates the SKEYSEED according to RFC7296

[9] The AES GCM IV is generated according to RFC5282

[10] The AES GCM IV is generated according to RFC4106

2.4 Disallowed Algorithms

These algorithms are non-Approved algorithms that are disabled when the module is operated in an Approved mode of operation.

- ARCFOUR
- Blowfish
- CAST
- DSA (SigGen, SigVer; non-compliant)
- HMAC-MD5
- HMAC-RIPEMD160
- UMAC

2.5 Critical Security Parameters

All CSPs and public keys used by the module are described in this section.

Table 11 – Critical Security Parameters (CSPs)

Name	Description and usage
DRBG_Seed	Seed material used to seed or reseed the DRBG
DRBG_State	V and Key values for the HMAC_DRBG
Entropy Input String	256 bits entropy (min) input used to instantiate the DRBG
SSH PHK	SSH Private host key. 1st time SSH is configured, the keys are generated. ECDSA P-256. Used to identify the host.
SSH DH	SSH Diffie-Hellman private component. Ephemeral Diffie-Hellman private key used in SSH. DH (N = 256 bit[11]), ECDH P-256, or ECDH P-384
SSH-SEKs	SSH Session Keys: SSH Session Encryption Key: TDES (3key) or AES; SSH Session Integrity Key: HMAC
ESP-SEKs	IPSec ESP Session Keys: IKE Session Encryption Key: TDES (3key) or AES; IKE Session Integrity Key: HMAC.
IKE-PSK	Pre-Shared Key used to authenticate IKE connections.
IKE-Priv	IKE Private Key. RSA 2048, RSA 4096, ECDSA P-256, or ECDSA P-384
IKE-SKEYID	IKE SKEYID. IKE secret used to derive IKE and IPsec ESP session keys.
IKE-SEKs	IKE Session Keys: IKE Session Encryption Key: TDES (3key) or AES; IKE Session Integrity Key: HMAC
IKE-DH-PRI	IKE Diffie-Hellman private component. Ephemeral Diffie-Hellman private key used in IKE. DH N = 224 bit, ECDH P-256, or ECDH P-384
CO-PW	ASCII Text used to authenticate the CO.
User-PW	ASCII Text used to authenticate the User.

[11] SSH generates a Diffie-Hellman private key that is 2x the bit length of the longest symmetric or MAC key negotiated.

Table 12 – Public Keys

Name	Description and usage
SSH-PUB	SSH Public Host Key used to identify the host. ECDSA P-256.
SSH-DH-PUB	Diffie-Hellman public component. Ephemeral Diffie-Hellman public key used in SSH key establishment. DH (L = 2048 bit,), ECDH P-256, or ECDH P-384
IKE-PUB	IKE Public Key RSA 2048, RSA 4096, ECDSA P-256, or ECDSA P-384
IKE-DH-PUB	Diffie-Hellman public component. Ephemeral Diffie-Hellman public key used in IKE key establishment. DH L = 2048 bit, ECDH P-256, or ECDH P-384
Auth-UPub	User Authentication Public Keys. Used to authenticate users to the module. ECDSA P256 or P-384
Auth-COPub	CO Authentication Public Keys. Used to authenticate CO to the module. ECDSA P256 or P-384
Root-CA	JuniperRootCA. ECDSA P-256 or P-384 X.509 Certificate; Used to verify the validity of the Juniper Package-CA at software load.
Package-CA	PackageCA. ECDSA P-256 X.509 Certificate; Used to verify the validity of Juniper Images at software load and also at runtime integrity.

3 Roles, Authentication and Services

3.1 Roles and Authentication of Operators to Roles

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

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

The User role monitors the router via the console or SSH. The user role may not change the configuration.

3.2 Authentication Methods

The module implements two forms of Identity-Based authentication, Username and password over the Console and SSH as well as Username and public key over SSH.

Password authentication: The module enforces 10-character passwords (at minimum) chosen from the 96 human readable ASCII characters. The maximum password length is 20-characters.

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. 4^{th} failed attempt = 10-second delay, 5^{th} failed attempt = 15-second delay, 6^{th} failed attempt = 20-second delay, 7^{th} failed attempt = 25-second delay).

This leads to a maximum of nine (9) 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. Thus, the probability of a successful random attempt is $1/96^{10}$, which is less than 1/1 million. The probability of a success with multiple consecutive attempts in a one-minute period is $9/(96^{10})$, which is less than 1/100,000.

ECDSA signature verification: SSH public-key authentication. Processing constraints allow for a maximum of 5.6e7 ECDSA attempts per minute. The module supports ECDSA (P-256 and P-384). The probability of a success with multiple consecutive attempts in a one-minute period is 5.6e7/ (2^{128}).

3.3 Services

All services implemented by the module are listed in the tables below. Table 15 lists the access to CSPs by each service.

Table 13 – Authenticated Services

Service	Description	CO	User
Configure security	Security relevant configuration	X	
Configure	Non-security relevant configuration	X	
Secure Traffic	IPsec protected connection (ESP)	X	
Status	Show status	X	x
Zeroize	Destroy all CSPs	X	

SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	X	x
IPsec connect	Initiate IPsec connection (IKE)	X	
Console access	Console monitoring and control (CLI)	X	x
Remote reset	Software initiated reset	X	

Table 14 – Unauthenticated traffic

Service	Description
Local reset	Hardware reset or power cycle
Traffic	Traffic requiring no cryptographic services

Table 15 – CSP Access Rights within Services

		CSPs												
Service	DRBG_Seed	DRBG_State	Entropy Input String	SSH PHK	SSH DH	SSH-SEK	ESP-SEK	IKE-PSK	IKE-Priv	IKE-SKEYID	IKE-SEK	IKE-DH-PRI	CO-PW	User-PW
Configure security	--	E	--	GWR	--	--	--	WR	GWR	--	--	--	W	W
Configure	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Secure traffic	--	--	--	--	--	--	E	--	--	--	E	--	--	--
Status	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Zeroize	Z	Z	Z	Z	Z	Z	Z	Z	Z	--	--	--	Z	Z
SSH connect	--	E	--	E	GE	GE	--	--	--	--	--	--	E	E
IPsec connect	--	E	--	--	--	--	G	E	E	GE	G	GE	--	--
Console access	--	--	--	--	--	--	--	--	--	--	--	--	E	E
Remote reset	GEZ	GZ	GZ	--	Z	Z	Z	--	--	Z	Z	Z	Z	Z
Local reset	GEZ	GZ	GZ	--	Z	Z	Z	--	--	Z	Z	Z	Z	Z
Traffic	--	--	--	--	--	--	--	--	--	--	--	--	--	--

G = Generate: The module generates the CSP
R = Read: The CSP is read from the module (e.g. the CSP is output)
E = Execute: The module executes using the CSP
W = Write: The CSP is updated or written to the module
Z = Zeroize: The module zeroizes the CSP.

3.4 Non-Approved Services

The following services are available in the non-Approved mode of operation. The security functions provided by the non-Approved services are identical to the Approved counterparts with the exception of SSH Connect (non-compliant) and IPsec Connect (non-compliant). SSH Connect (non-compliant) supports the security functions identified in Section 2.4 and the SSHv2 row of Table 10. The IPsec (non-compliant) supports the DSA in Section 2.4 and the IKEv1, IKEv2 and IPSec rows of Table 10.

Table 16 – Authenticated Services

Service	Description	CO	User
Configure security (non-compliant)	Security relevant configuration	X	
Configure (non-compliant)	Non-security relevant configuration	X	
Secure Traffic (non-compliant)	IPsec protected connection (ESP)	X	
Status (non-compliant)	Show status	X	x
Zeroize (non-compliant)	Destroy all CSPs	X	
SSH connect (non-compliant)	Initiate SSH connection for SSH monitoring and control (CLI)	X	x
IPsec connect (non-compliant)	Initiate IPsec connection (IKE)	X	
Console access (non-compliant)	Console monitoring and control (CLI)	X	x
Remote reset (non-compliant)	Software initiated reset	X	

Table 17 – Unauthenticated traffic

Service	Description
Local reset (non-compliant)	Hardware reset or power cycle
Traffic (non-compliant)	Traffic requiring no cryptographic services

4 Self-tests

Each time the module is powered up, it tests that the cryptographic algorithms still operate correctly and that sensitive data have not been damaged. Power-up self–tests are available on demand by power cycling the module.

On power-up or reset, the module performs the self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fails, the module enters the Critical Failure error state.

The module performs the following power-up self-tests:

- Software Integrity check using ECDSA P-256 with SHA-256
- **Data Plane KATs**
    - AES-CBC (128/192/256) Encrypt KAT
    - AES-CBC (128/192/256) Decrypt KAT
    - Triple-DES-CBC Encrypt KAT
    - Triple-DES-CBC Decrypt KAT
    - HMAC-SHA-1 KAT
    - HMAC-SHA-256 KAT
    - AES-GCM (128/192/256) Encrypt KAT
    - AES-GCM (128/192/256) Decrypt KAT
- **Control Plane QuickSec KATs**
    - SP 800-90A HMAC DRBG KAT
        - Health-tests initialize, re-seed, and generate
    - RSA 2048 w/ SHA-256 Sign KAT
    - RSA 2048 w/ SHA-256 Verify KAT
    - ECDSA P-256 w/ SHA-256 Sign/Verify PCT
    - Triple-DES-CBC Encrypt KAT
    - Triple-DES-CBC Decrypt KAT
    - HMAC-SHA-256 KAT
    - HMAC-SHA-384 KAT
    - AES-CBC (128/192/256) Encrypt KAT
    - AES-CBC (128/192/256) Decrypt KAT
    - AES-GCM (128/256) Encrypt KAT
    - AES-GCM (128/256) Decrypt KAT
    - KDF-IKE-V1 KAT
    - KDF-IKE-V2 KAT
- **OpenSSL KATs**
    - SP 800-90A HMAC DRBG KAT
        - Health-tests initialize, re-seed, and generate.
    - ECDSA P-256 Sign/Verify PCT
    - ECDH P-256 KAT
        - Derivation of the expected shared secret.
    - RSA 2048 w/ SHA-256 Sign KAT
    - RSA 2048 w/ SHA-256 Verify KAT
    - Triple-DES-CBC Encrypt KAT
    - Triple-DES-CBC Decrypt KAT

- HMAC-SHA-1 KAT
    - HMAC-SHA-256 KAT
    - HMAC-SHA-384 KAT
    - HMAC-SHA-512 KAT
    - AES-CBC (128/192/256) Encrypt KAT
    - AES-CBC (128/192/256) Decrypt KAT
- **OpenSSH KATs**
    - KDF-SSH KAT
- **LibMD KATs**
    - SHA-256
    - SHA-512
- Critical Function Test
    - The cryptographic module performs a verification of a limited operational environment, and verification of optional non-critical packages.

The module also performs the following conditional self-tests:

- Continuous RNG Test on the SP 800-90A HMAC-DRBG
- Continuous RNG test on the NDRNG
- Pairwise consistency test when generating ECDSA, and RSA key pairs.
- Software Load Test (ECDSA signature verification)

5 Physical Security Policy

The module's physical security requirements do not apply to the Juniper Networks vSRX Virtual Firewall because the module is a FIPS 140-2 Level 1 software module and the physical security is provided by the host platform.

6 Security Rules and Guidance

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. Power up self-tests 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 requires two independent internal actions to be performed prior to outputting plaintext CSPs.
11. The cryptographic officer must determine whether software being loaded is a legacy use of the software load service.
12. The cryptographic officer must retain control of the module while zeroization is in process.
13. 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.
14. The cryptographic officer must configure the module to IPsec ESP lifetime-kilobytes to ensure the module does not encrypt more than 2^32 blocks with a single Triple-DES key when Triple-DES is the encryption-algorithm for IKE and/or IPsec ESP.

6.1 Crypto-Officer Guidance

The crypto-officer is responsible for installing the module on the platform on which the module was tested and validated, configuring the module in FIPS mode and configuring the operator's usernames and passwords.

Guide to Download Software Packages for vSRX from Juniper Networks:

1. Using a Web browser, follow the link to the download URL on the Juniper Networks webpage at http://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: 15.1X49).
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”
    - Verify the checksum of the download with the provided checksum

The crypto-officer shall follow the instructions for installation provided in the Juniper Networks *vSRX Guide for VMware* documentation. Once the FIPS 140-2 validated vSRX *software* is installed on the hardware platform and hypervisor in Table 1 then the crypto-officer shall follow the instructions in section 1.2 of the security policy to place the module in the FIPS Approved mode of operation.

7 References and Definitions

The following standards are referred to in this Security Policy.

Table 18 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Module*, May 25, 2001
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011
[IG]	*Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*
[135]	*National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011.*
[186]	National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July, 2013.
[197]	*National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001*
[38A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001*
[38D]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Special Publication 800-38D, November 2007*
[198]	*National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008*
[180]	*National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015*
[67]	*National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, May 2004*
[90A]	National Institute of Standards and Technology, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, June 2015.

Table 19 – Acronyms and Definitions

Acronym	Definition
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange Protocol
IPsec	Internet Protocol Security
MD5	Message Digest 5
RSA	Public-key encryption technology developed by RSA Data Security, Inc.
SHA	Secure Hash Algorithms
SSH	Secure Shell
Triple-DES	Triple - Data Encryption Standard

Table 20 – Datasheets

Model	Title	URL
vSRX	vSRX Virtual Firewall	http://www.juniper.net/assets/us/en/local/pdf/datasheets/1000489-en.pdf