Juniper Networks SRX300, SRX320, SRX340, SRX345 and SRX550-M Services Gateways

Non-Proprietary FIPS 140-2 Cryptographic Module Level One Security Policy

Version: 1.8

Date: December 22, 2017

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 SRX Series Services Gateways are a series of secure routers 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. All models run Juniper's JUNOS firmware – in this case, a specific FIPS-compliant version, when configured in FIPS-MODE called JUNOS-FIPS-MODE, version 15.1X49-D60. The firmware image is junos-srxsme-15.1X49-D60.10-domestic.tgz and the firmware Status service identifies itself as in the "Junos OS 15.1X49-D60.10".

This Security Policy covers the "Branch" models – the SRX300, SRX320, SRX340, SRX345 and SRX550-M[1] models. They are meant for corporate branch offices of various sizes. (Intended size is proportional to model number.)

The cryptographic modules are defined as multiple-chip standalone modules that execute JUNOS firmware on any of the Juniper Networks SRX Series Services Gateways listed in the table below.

Table 1 – Cryptographic Module Configurations

Model	Hardware Versions	Firmware	Distinguishing Features
SRX300	SRX300	JUNOS 15.1X49-D60	6 x 10/100/1000; 2 SFP
SRX320	SRX320	JUNOS 15.1X49-D60	86 x 10/100/1000; 2 SFP; 2 MPIM expansion slots
SRX340	SRX340	JUNOS 15.1X49-D60	8 x 10/100/1000; 4 SFP; 4 MPIM expansion slots; 1 x 10/100/1000 management port
SRX345	SRX345	JUNOS 15.1X49-D60	8 x 10/100/1000; 4 SFP; 4 MPIM expansion slots; 1 x 10/100/1000 management port
SRX550-M[1]	SRX550-645AP-M SRX550-645DP-M	JUNOS 15.1X49-D60	6 x 10/100/1000; 4 SFP; 2 MPIM expansion slots; 6 GPIM expansion slots

[1] SRX550-M and SRX550 refer to the same security appliance model, whereas SRX550-M is convention used in FIPS Security Policy as well as other Juniper public documents and SRX550 is convention for model name displayed on the actual hardware faceplate.

The modules are 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	1
6	Operational Environment	N/A
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 modules have a limited operational environment as per the FIPS 140-2 definitions. They include a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into these modules is out of the scope of this validation and require a separate FIPS 140-2 validation.

The modules do not implement any mitigations of other attacks as defined by FIPS 140-2.

1.1 Hardware and Physical Cryptographic Boundary

The physical forms of the module's various models are depicted in Figures 1-5 below. For all models the cryptographic boundary is defined as the outer edge of the chassis, but for the SRX550-M the power supplies are excluded.

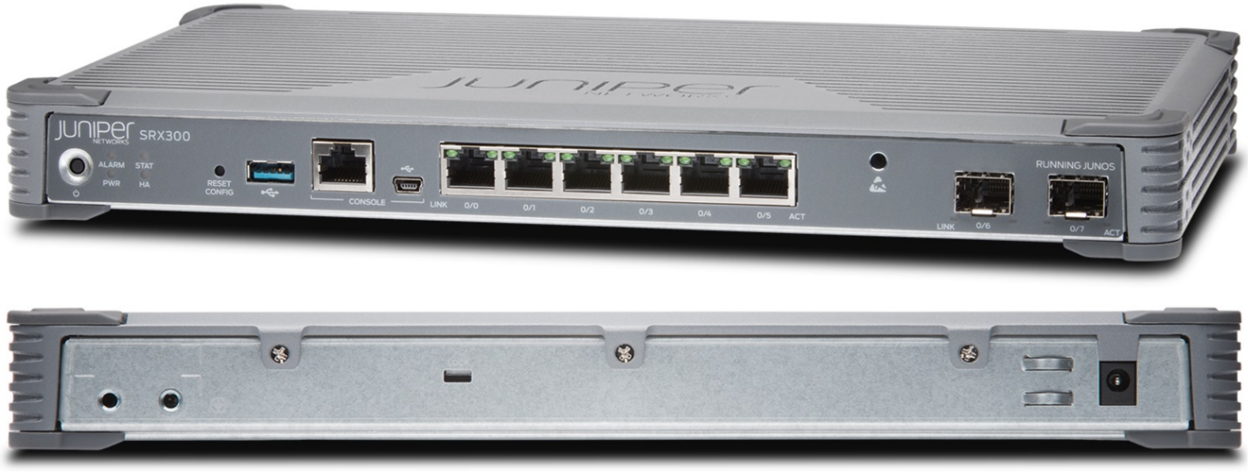

Figure 1. SRX 300

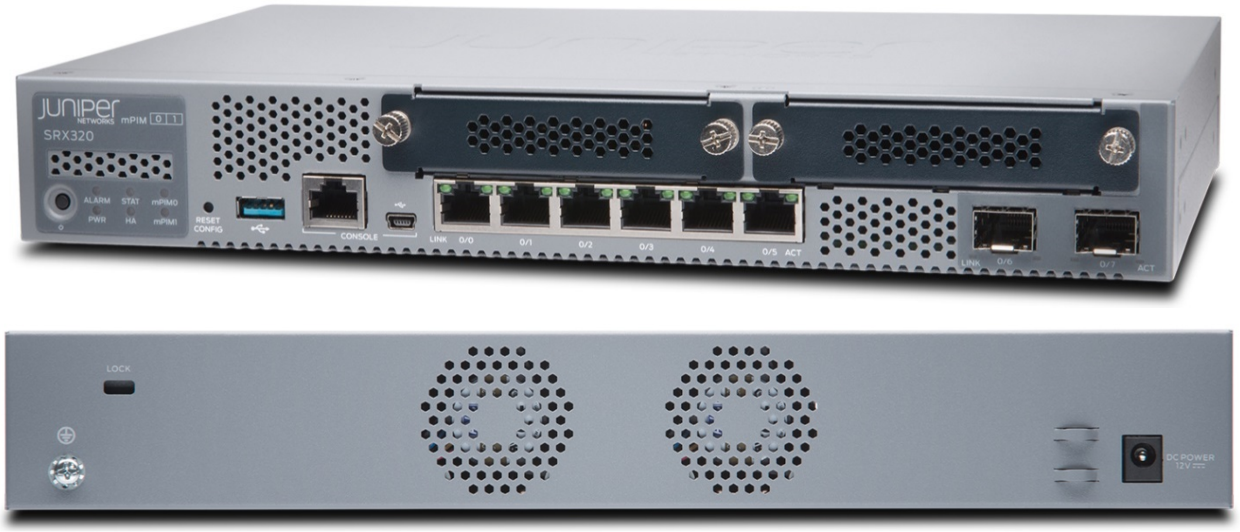

Figure 2. SRX 320

Figure 3. SRX 340

Figure 4. SRX 345

Figure 5. SRX 550-M

Table 3 - Ports and Interfaces

Port	Description	Logical Interface Type
Ethernet	LAN Communications	Control in, Data in, Data out, Status out
Serial	Console serial port	Control in, Status out
Power	Power connector	Power in
Reset	Reset button	Control in
LED	Status indicator lighting	Status out
USB	Firmware load port	Control in, Data in
WAN	SHDSL, VDSL, T1, E1	Control in, Data in, Data out, Status out

1.2 Mode of Operation

The cryptographic module provides a non-Approved mode of operation in which non-Approved cryptographic algorithms are supported. The module supports non-Approved algorithms when operating in the non-Approved mode of operation as described in Sections 2.4 and 3.4. When transitioning between the non-Approved mode of operation and the Approved mode of operation, the CO must zeroize all CSPs by following the instructions in Section 1.3.

Then, the CO must run the following commands to configure the module into the Approved mode of operation:

```
co@fips-srx# set system fips level 2

co@fips-srx# commit
```

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

```
co@fips-srx:fips# set security ike gateway <name> version v2-only
```

<name> - the user configured name for the IKE gateway

```
co@fips-srx:fips# commit
```

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 operator can verify the module is operating in the Approved mode by verifying the following:

- The "show version" command indicates that the module is running the Approved firmware (i.e. JUNOS Software Release [15.1X49-D60]).
- The command prompt ends in ":fips", which indicates the module has been configured in the Approved mode of operation.

- The “show security ike” and “show security ipsec” commands show IKEv2 is configured when either an IPsec or IKE proposal is configured to use AES-GCM.

1.3 Zeroization

The following command allows the Cryptographic Officer to zeroize CSPs contained within the module:

```
co@fips-srx> request system zeroize
```

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

2. Cryptographic Functionality

The module implements the FIPS Approved, vendor affirmed, and non-Approved but Allowed cryptographic functions listed in Table 4 through Table 10 below. Table 11 summarizes the high level protocol algorithm support.

2.1 Approved Algorithms

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

Items enclosed in curly brackets { } are CAVP tested but not used by the module in the Approved mode.

Note: Some Data Plane and Control Plane implementations have three algorithm certificates due to hardware accelerated implementations. Each model utilizes one of the three algorithm certificates, depending on the hardware.

Table 4 – Data Plane Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
4348 4347 4346	AES [197]	CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		GCM [38D]	Key Sizes: 128, 192, 256	Encrypt, Decrypt, AEAD
2888 2887 2886	HMAC [198]	SHA-1	λ = 96	Message Authentication
		SHA-256	λ = 128	
3585 3584 3583	SHS [180]	SHA-1 SHA-256		Message Digest Generation
2352 2351 2350	Triple-DES [67]	TCBC [38A]	Key Size: 3-key (3 independent 64-bit keys)	Encrypt, Decrypt

Table 5 – Control Plane Authentec Approved Cryptographic Functions

Cert	Algorithm	Mode	Description	Functions
4345	AES [197]	CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		GCM [38D]	Key Sizes: 128, 256	Encrypt, Decrypt, AEAD
N/A[2]	CKG	[133] Section 6.2		Asymmetric key generation using unmodified DRBG output
		[133] Section 7.3		Derivation of symmetric keys
1051	CVL	IKEv1 [135]	SHA {1}, 256, 384	Key Derivation
		IKEv2 [135]	SHA {1}, 256, 384	
1041 1040 1039	ECDSA [186]		P-256 (SHA 256) P-384 (SHA {256}, 384)	KeyGen, SigGen, SigVer
2885	HMAC [198]	{SHA-1}		IKE Message Authentication, IKE KDF Primitive
		SHA-256	λ = 128, 256	
		SHA-384	λ = 192, 384	
N/A	KTS	AES Cert. #4345 and HMAC Cert. #2885		Key establishment methodology provides between 128 and 256 bits of encryption strength
		Triple-DES Cert. #2349 and HMAC Cert. #2885		Key establishment methodology provides 112 bits of encryption strength

2361
2360
2359
	RSA [186]	PKCS1_V1_5	n=2048 (SHA 256)
			n=4096 (SHA 256) [3]
				SigGen, SigVer

3582	SHS [180]	{SHA-1} SHA-256 SHA-384		Message Digest Generation
2349	Triple-DES [67]	TCBC [38A]	Key Size: 3-key (3 independent 64-bit keys)	Encrypt, Decrypt

[2] Vendor Affirmed.

[3] RSA 4096 SigGen was tested to FIPS 186-4; however, the CAVP certificate lists 4096 under FIPS 186-2.

Table 6 – OpenSSL Approved Cryptographic Functions

Cert	Algorithm	Mode	Description	Functions
1398	DRBG [90A]	HMAC	SHA-256	Control Plane Random Bit Generation/ OpenSSL Random Bit Generator

Table 7 – OpenSSL Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Description	Functions
4362	AES [197]	CBC [38A] CTR[38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
N/A[4]	CKG	[133] Section 6.1 [133] Section 6.2		Asymmetric key generation using unmodified DRBG output
		[133] Section 7.3		Derivation of symmetric keys

1038	ECDSA [186]		{P-224 (SHA 256)}
			P-256 (SHA 256)
			{P-384 (SHA 256)}
				SigGen

			{P-224 (SHA 256)}
			P-256 (SHA 256)
			P-384 (SHA {256}, 384)
			{P-521 (SHA-256)}
				KeyGen, SigVer

2902	HMAC [198]	SHA-1	λ = 160	SSH Message Authentication DRBG Primitive
		SHA-256	λ = 256	
		SHA-512	λ = 512	
N/A	KTS	AES Cert. #4362 and HMAC Cert. #2902		Key establishment methodology provides between 128 and 256 bits of encryption strength
		Triple-DES Cert. #2358 and HMAC Cert. #2902		Key establishment methodology provides 112 bits of encryption strength

2358	RSA [186]		n=2048 (SHA 256)
			{n=3072 (SHA 256)}
			n=4096 (SHA 256)[5]
				SigGen

			n=2048 (SHA 256)
			{n=3072 (SHA 256)}
				KeyGen, SigVer

3600	SHS [180]	SHA-1 SHA-256 SHA-384		Message Digest Generation, SSH KDF Primitive
		SHA-512		Message Digest Generation
2358	Triple-DES [67]	TCBC [38A]	Key Size: 3-key (3 independent 64-bit keys)	Encrypt, Decrypt

[4] Vendor Affirmed.

[5] RSA 4096 SigGen was tested to FIPS 186-4; however, the CAVP certificate lists 4096 under FIPS 186-2.

Table 8 – OpenSSH Approved Cryptographic Functions

Cert	Algorithm	Mode	Description	Functions
1071	CVL	SSH [135]	SHA 1, 256, 384	Key Derivation

Table 9 – LibMD Approved Cryptographic Functions

Cert	Algorithm	Mode	Description	Functions
3586	SHS [180]	SHA-256 SHA-512		Message Digest Generation

2.2 Allowed Algorithms

Table 10 - 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	Provides a minimum of 256 bits of entropy.	Seeding the DRBG

2.3 Allowed Protocols

Table 11 - Protocols Allowed in FIPS Mode

Protocol	Key Exchange	Auth	Cipher	Integrity
IKEv1	Diffie-Hellman (L = 2048, N = 2047) EC Diffie-Hellman P-256, P-384	RSA 2048 RSA 4096 Pre-Shared Secret ECDSA P-256 ECDSA P-384	Triple-DES CBC[6] AES CBC 128/192/256	SHA-256,384
IKEv2[7]	Diffie-Hellman (L = 2048, N = 2047) EC Diffie-Hellman P-256, P-384	RSA 2048 RSA 4096 Pre-Shared Secret ECDSA P-256 ECDSA P-384	Triple-DES CBC[8] AES CBC 128/192/256 AES GCM[9] 128/256	SHA-256,384
IPsec ESP	IKEv1 with optional: • Diffie-Hellman (L = 2048, N = 2047) • EC Diffie-Hellman P-256, P-384	IKEv1	3 Key Triple-DES CBC[10] AES CBC 128/192/256	HMAC-SHA-1-96 HMAC-SHA-256-128
	IKEv2 with optional: • Diffie-Hellman (L = 2048, N = 2047) • EC Diffie-Hellman P-256, P-384	IKEv2	3 Key Triple-DES CBC[11] AES CBC 128/192/256 AES GCM[12] 128/192/256	
SSHv2	Diffie-Hellman (L = 2048, N = 2047) EC Diffie-Hellman P-256, P-384	ECDSA P-256	Triple-DES CBC[13] AES CBC 128/192/256 AES CTR 128/192/256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

No parts of the IKEv1, IKEv2, ESP, and SSHv2 protocols, other than the KDF, have been tested by the CAVP or CMVP.

The IKE and SSH algorithms allow independent selection of key exchange, authentication, cipher, and integrity. In **Table 11 - Protocols Allowed in FIPS Mode** 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.

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

[6] The Triple-DES key for the IETF IKEv1 protocol is generated according to RFC 2409.
[7] IKEv2 generates the SKEYSEED according to RFC7296.
[8] The Triple-DES key for the IETF IKEv2 protocol is generated according to RFC 7296.
[9] The GCM IV is generated according to RFC5282.
[10] The Triple-DES key for the ESP protocol is generated by the IETF IKEv1 protocol according to RFC 2409.
[11] The Triple-DES key for the ESP protocol is generated by the IETF IKEv2 protocol according to RFC 7296.
[12] The GCM IV is generated according to RFC4106.
[13] The Triple-DES key for the IETF SSHv2 protocol is generated according to RFCs 4253 and 4344.

- 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 12 - Critical Security Parameters (CSPs)

Name	Description and usage	CKG
DRBG_Seed	Seed material used to seed or reseed the DRBG	N/A
DRBG_State	V and Key values for the HMAC_DRBG	N/A
SSH PHK	SSH Private host key. 1st time SSH is configured, the keys are generated. ECDSA P-256. Used to identify the host.	[133] Section 6.1
SSH DH	SSH Diffie-Hellman private component. Ephemeral Diffie-Hellman private key used in SSH. Diffie-Hellman (N = 256 bit, 320 bit, 384 bit, 512 bit, or 1024 bit[14][15]), EC Diffie-Hellman P-256, or EC Diffie-Hellman P-384	[133] Section 6.2
SSH-SEK	SSH Session Key; Session keys used with SSH. Triple-DES (3key), AES, HMAC.	[133] Section 7.3
ESP-SEK	IPsec ESP Session Keys. Triple-DES (3 key), AES, HMAC.	[133] Section 7.3
IKE-PSK	Pre-Shared Key used to authenticate IKE connections.	N/A
IKE-Priv	IKE Private Key. RSA 2048, RSA 4096, ECDSA P-256, or ECDSA P-384	[133] Section 6.1
IKE-SKEYID	IKE SKEYID. IKE secret used to derive IKE and IPsec ESP session keys.	[133] Section 7.3
IKE-SEK	IKE Session Keys. Triple-DES (3 key), AES, HMAC.	[133] Section 7.3
IKE-DH-PRI	IKE Diffie-Hellman private component. Ephemeral Diffie-Hellman private key used in IKE. Diffie-Hellman N = 224 bit, EC Diffie-Hellman P-256, or EC Diffie-Hellman P-384	[133] Section 6.2
CO-PW	ASCII Text used to authenticate the CO.	N/A
User-PW	ASCII Text used to authenticate the User.	N/A

Table 13 - Public Keys

Name	Description and usage	CKG
SSH-PUB	SSH Public Host Key used to identify the host. ECDSA P-256.	[133] Section 6.1
SSH-DH-PUB	Diffie-Hellman public component. Ephemeral Diffie-Hellman public key used in SSH key establishment. DH (L = 2048 bit), EC Diffie-Hellman P-256, or EC Diffie-Hellman P-384	[133] Section 6.2
IKE-PUB	IKE Public Key RSA 2048, RSA 4096, ECDSA P-256, or ECDSA P-384	[133] Section 6.1

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

.

Name	Description and usage	CKG
IKE-DH-PUB	Diffie-Hellman public component. Ephemeral Diffie-Hellman public key used in IKE key establishment. Diffie-Hellman L = 2048 bit, EC Diffie-Hellman P-256, or EC Diffie-Hellman P-384	[133] Section 6.2
Auth-UPub	SSH User Authentication Public Keys. Used to authenticate users to the module. ECDSA P-256 or P-384	N/A
Auth-COPub	SSH CO Authentication Public Keys. Used to authenticate CO to the module. ECDSA P-256 or P-384	N/A
Root CA	Juniper Root CA. ECDSA P-256 or P-384 X.509 Certificate; Used to verify the validity of the Juniper Package CA at software load.	N/A
Package CA	Package CA. ECDSA P-256 X.509 Certificate; Used to verify the validity of Juniper Images at software load and boot.	N/A

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. 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 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; 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 56,000,000 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 $56{,}000{,}000/(2^{128})$, which is less than 1/100,000.

3.3 Services

All services implemented by the module are listed in the tables below. Table 16 lists the access to CSPs by each service, and Table 17 lists the access to Public Keys by each service.

Table 14 - 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 15 - Unauthenticated Traffic

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

Table 16 - CSP Access Rights within Services

Service	CSPs												
	DRBG_Seed	DRBG_State	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
SSH connect	--	E	E	GE	GE	--	--	--	--	--	--	E	E
IPsec connect	--	E	--	--	--	G	E	E	GE	G	GE	--	--
Console access	--	--	--	--	--	--	--	--	--	--	--	E	E
Remote reset	GZE	GZ	--	Z	Z	Z	--	--	Z	Z	Z	Z	Z
Local reset	GZE	GZ	--	Z	Z	Z	--	--	Z	Z	Z	Z	Z

Service	CSPs												
	DRBG_Seed	DRBG_State	SSH PHK	SSH DH	SSH-SEK	ESP-SEK	IKE-PSK	IKE-Priv	IKE-SKEYID	IKE-SEK	IKE-DH-PRI	CO-PW	User-PW
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 written to persistent storage in the module.
Z = Zeroize: The module zeroizes the CSP.

Table 17: Public Key Access Rights within Services

Service	Public Keys							
	SSH-PUB	SSH-DH-PUB	IKE-PUB	IKE-DH-PUB	Auth-UPub	Auth-COPub	Root-CA	Package-CA
Configure security	GWR	-	GWR	-	W	W	-	-
Configure	-	-	-	-	-	-	-	-
Secure traffic	-	-	-	-	-	-	-	-
Status	-	-	-	-	-	-	-	-
Zeroize	Z	-	Z	Z	Z	Z	-	-
SSH connect	E	GE	-	-	E	E	-	-
IPsec connect	-	-	E	GE	-	-	-	-
Console access	-	-	-	-	-	-	-	-
Remote reset	-	Z	-	Z	Z	Z	-	E
Local reset	-	Z	-	Z	Z	Z	-	E
Traffic	-	-	-	-	-	-	-	-
Software load	-	-	-	-	-	-	EW	EW

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

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). SSH Connect (non-compliant) supports the security functions identified in Section 2.3 and the SSHv2 row of Table 11.

Table 18 - 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 19 - 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 has 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:

- Firmware 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 Authentec KATs**
  - 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-SHA2-256 KAT
  - HMAC-SHA2-384 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
  - EC Diffie-Hellman 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-SHA2-256 KAT
  - HMAC-SHA2-384 KAT (used to satisfy the SHA-384 KAT)
  - HMAC-SHA2-512 KAT

  - AES-CBC (128/192/256) Encrypt KAT
  - AES-CBC (128/192/256) Decrypt KAT
- **OpenSSH KAT**
  - KDF-SSH KAT
- **Libmd KATs**
  - HMAC-SHA2-256 KAT (used to satisfy the SHA-256 KAT)
  - SHA-2-512 KAT

- Critical Function Test
  - The cryptographic module performs a verification of a limited operational environment.

Upon successful completion of the self-tests, the module outputs “FIPS self-tests completed.” to the local console.

If a self-test fails, the module outputs “<self-test name>: Failed” to the local console and automatically reboots.

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.
- Firmware Load Test (ECDSA P-256 with SHA-256 signature verification)

5. 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 (i.e. SSH PHK, IKE-PSK, or IKE-Priv via the Configure security service).
11. The cryptographic officer must determine whether firmware being loaded is a legacy use of the firmware load service.
12. The cryptographic officer must retain control of the module while zeroization is in process.
13. The cryptographic officer must configure the module to use IKEv2 when GCM is configured for IKE or IPsec ESP.
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. References and Definitions

The following standards are referred to in this Security Policy.

Table 20 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, 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*
[133]	*NIST Special Publication 800-133, Recommendation for Cryptographic Key Generation, December 2012*
[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.*
[186-2]	*National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-2, January 2000.*
[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 21 – Acronyms and Definitions

Acronym	Definition
AES	Advanced Encryption Standard
DSA	Digital Signature Algorithm
EC Diffie-Hellman	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm

Acronym	Definition
EMC	Electromagnetic Compatibility
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
GPIM	Gigabit-Backplane Pluggable Interface Module
HMAC	Keyed-Hash Message Authentication Code
ICV	Integrity Check Value (i.e. Tag)
IKE	Internet Key Exchange Protocol
IOC	Input/Output Card
IPsec	Internet Protocol Security
MD5	Message Digest 5
MPIM	Mini-Pluggable Interface Module
NPC	Network Processing Card
PIM	Pluggable Interface Module
RE	Routing Engine
RSA	Public-key encryption technology developed by RSA Data Security, Inc.
SHA	Secure Hash Algorithms
SPC	Services Processing Card
SSH	Secure Shell
Triple-DES	Triple - Data Encryption Standard

Table 22 – Datasheets

Model	Title	URL
SRX300 SRX320 SRX340 SRX345	SRX300 Line of Services Gateways for the Branch	http://www.juniper.net/assets/us/en/local/pdf/datasheets/1000550-en.pdf
SRX550-M	SRX Series Services Gateways for the Branch	http://www.juniper.net/assets/us/en/local/pdf/datasheets/1000281-en.pdf