Security Target

Juniper Networks EX4600 and QFX5100 Switches Running Junos OS 14.1X53-D30

ST Version 1.0

December 10, 2015

Prepared By:

Juniper Networks, Inc.

1133 Innovation Way

Sunnyvale, CA 94089

www.juniper.net

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Junos OS 14.1X53-D30 running on EX4600 and QFX5100 Ethernet Switches. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

Table of Contents

List of Tables

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), Security Target organization, document conventions, and terminology. It also includes an overview of the evaluated products.

1.1 ST Reference

ST Title	Security Target: Juniper Networks EX4600 and QFX5100 Switches Running Junos OS 14.1X53-D30
ST Revision	1.0
ST Draft Date	December 10, 2015
Author	Juniper Networks, Inc.

1.2 TOE Reference

TOE Reference	Juniper Networks EX4600 and QFX5100 Switches Running Junos OS 14.1X53-D30

1.3 About This Document

This Security Target follows the following format:

SECTION	TITLE	DESCRIPTION
1	Introduction	Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable
3	Security Problem Definition	Specifies the threats, assumptions and organizational security policies that affect the TOE
4	Security Objectives	Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives satisfy the threats
5	Security Requirements	Contains the functional and assurance requirements for this TOE
6	TOE Summary Specification	Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements
7	Rationale	Demonstrates traceability and internal consistency
8	Audit Events	TOE audit events are listed here
9	Appendices	Supporting material

Table 1 - ST Organization and Section Descriptions

1.3.1 Document Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC that are not already completed in [NDPP][1]:

- Assignment: Indicated with *italicized* text;
- Refinement made by ST author: Indicated with **bold** text and strikethroughs, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined* text.

Iterations are indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3). Iterations identified in [NDPP] are identified in the same manner in this ST.

1.3.2 Document Terminology

See Section 11.2 for the Glossary.

1.4 TOE Overview

The Target of Evaluation (TOE) is a network device (switch), and includes the following secure network devices running Junos OS 14.1X53-D30

- EX4600
- QFX5100

1.5 TOE Boundaries

The TOE consists of the following IT components:

1. Network devices (as detailed in Table 2 below).
2. Junos OS 14.1X53-D30 package: incorporating the Junos OS 14.1X53 operating system for security switching appliances and the CentOS 6.4 providing full hardware virtualization.

The TOE is managed and configured via the Junos OS Command Line Interface.

The EX-series and QFX-series switches provide high-performance, carrier-class networking solutions, supporting a variety of high-speed Ethernet interfaces for medium/large networks.

The hardware has two components: the switch chassis and the Small Form-factor Pluggable (SFP) interfaces that have been installed in the switch. The various SPFs that have been installed in switch allow it to communicate with the different types of networks that may be required within the environment where the switch will be used[2]. These are detailed in Section 10 of this ST.

[1] i.e. if a selection, assignment or refinement has been made in [NDPP] it will not also be marked using the font conventions (although any square brackets used in [NDPP] will be retained) in this security target, thereby highlighting the additional operations completed in the Security Target.

[2] These network interfaces are required for the TOE to operate. However, they are not relied upon for the enforcement security functionality necessary to satisfy the requirements of [NDPP] and so do not fall within the scope of the TSF. Therefore,

The software package is comprised of two components: the CentOS 6.4 kernel providing full hardware virtualization and the Junos OS 14.1X53 providing security switching. A combined install package is created of the Junos OS virtual machine (VM), together with the CentOS kernel.

CentOS 6.4 provides full hardware virtualization using *hardware-assisted virtualization*. This allows Junos OS to run on the virtualization platform as an unmodified guest operating systems. CentOS is responsible for presenting the (emulated) hardware devices to Junos OS, which Junos OS can then address as it would address any physical device.

The Junos OS consists of the following major architectural components:

- The Routing Engine (RE), which provides Layer 3 routing services and network management and control;
- The Packet Forwarding Engine (PFE)[3], which provides all operations necessary for transit packet forwarding.

The Routing Engine and Packet Forwarding Engine perform their primary tasks independently, while constantly communicating through a high-speed internal link. This arrangement provides streamlined forwarding and routing control and the capability to run Internet-scale networks at high speeds.

1.5.1 Physical Boundary

Series	Model	Ports[4]	Firmware[5]
EX-Series	EX4600	1GbE SFP: 24(40) (with 10GbE expansion modules) 10GbE SFP+: 24(40/72) (with 10GbE expansion modules/with fixed 40GbE ports using breakout cables) 40GbE QSFP+: 4(12) (with expansion modules)	Junos OS 14.1X53-D30.3
QFX-Series	QFX5100	100 Mbps RJ-45 1GbE RJ-45 10GbE RJ-45 1GbE SFP 10GbE SFP+ 40GbE QSFP+	Junos OS 14.1X53-D30.3

Table 2 - List of Network Device Hardware

The TOE is comprised of the Junos OS 14.1X53-D30 firmware together with the CentOS kernel (providing the virtualized environment in which Junos OS VM executes) running on the appliance

the network interfaces are considered to be non-TOE hardware/software/firmware entities, and are referenced as such in section 1.5.3.

[3] The network interface components form the lower layers of the PFE (the SPFs and Line Cards) which simply deal with physical interfaces mechanics.

[4] The SFP/line cards plugged into the chassis ports are considered to be non-TOE hardware/software/firmware entities as discussed above.

[5] The firmware version reflects the detail reported for the components of the Junos OS when the show version command is executed on the appliance.

chassis listed in Table 2 above (including the software implementing the Routing Engine and the software and ASICs implementing the Packet Forwarding Engine). Hence the TOE is contained within the physical boundary of the specified appliance chassis.

Details of the appliance specific install packages of the TOE firmware (Junos OS 14.1X53-D30 bundled with the CentOS 6.4 kernel) are provided in Section 9, Install Packages.

The guidance documents included as part of the TOE are:

[SG_EX]	Complete Software Guide for Junos OS for EX4600 Ethernet Switches, Release 14.1X53
[SG_QFX]	Complete Software Guide for Junos OS for QFX Series Switches, Release 14.1X53d30
[ECG14.1]	Junos OS Common Criteria Evaluation Configuration Guide for QFX5100 and EX4600 Devices Release 14.1X53D30

1.5.2 Logical Boundary

This section outlines the boundaries of the security functionality of the TOE; the logical boundary of the TOE includes the security functionality described in the following sections.

TSF	DESCRIPTION
Audit (FAU)	Junos auditable events are stored in the syslog files, and can be sent to an external log server (via Netconf over SSH). Auditable events include start-up and shutdown of the audit functions, authentication events, service requests, as well as the events listed in the table in Section 8. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local syslog storage limits are configurable and are monitored. In the event of storage limits being reached the oldest logs will be overwritten.
Cryptographic Support (FCS)	The TOE includes a baseline cryptographic module that provides confidentiality and integrity services for authentication and for protecting communications with adjacent systems.
User Data Protection (FDP)	The TOE is designed to process network packets and forward them as appropriate. The packet handling is implemented in such a manner as to prevent the leakage of user data from one packet into other packet(s) there were not intended by the originator.
Identification and Authentication (FIA)	The TOE requires users to provide unique identification and authentication data before any administrative access to the system is granted. The devices also require that applications exchanging information with them successfully authenticate prior to any exchange. This covers all services used to exchange information, including Secure Shell (SSH). Telnet, File Transfer Protocol (FTP), Secure Socket Layer (SSL) are out of scope.
Security Management (FMT)	The TOE provides an Authorized Administrator role that is responsible for: • the configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the evaluated product • the regular review of all audit data;

TSF	DESCRIPTION
	• all administrative tasks (e.g., creating the security policy). The devices are managed through a Command Line Interface (CLI). The CLI is accessible through remote administrative session.
Protection of the TSF (FPT)	The TOE provides protection mechanisms for its security functions. One of the protection mechanisms is to protect TFS data (e.g. cryptographic keys, administrator passwords). Another protection mechanism is to ensure the integrity of any software/firmware updates are can be verified prior to installation. The TOE provides for both cryptographic and non-cryptographic self-tests, and is capable of automated recovery from failure states. Also, reliable timestamp is made available for use by the TOE.
TOE Access (FTA)	The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.
Trusted Path/Channels (FTP)	The TOE creates trusted channels between itself and remote trusted authorized IT product (e.g. syslog server) entities that protect the confidentiality and integrity of communications. The TOE creates trusted paths between itself and remote administrators and users that protect the confidentiality and integrity of communications.

Table 3 - TOE Logical Boundary

1.5.3 Non-TOE hardware, software, firmware

Small Form-factor Pluggable (SFP)s are required by the TOE to operate, communicate with the connected network. These are detailed for each TOE appliance in Section 10.

The TOE requires the following clients/servers to be provided in the connected network:

- Syslog server supporting SSHv2 connections to send audit logs
- SSHv2 client for remote administration
- Serial connection client for local administration

1.5.4 Summary of Out-of-Scope Items

The only security functionality addressed by the evaluation is the functionality specified by the functional requirements in Section 6.1, and does not include additional product capabilities such as use of information flow control based on traffic filters. The following items are out of the scope of the evaluation:

- Use of telnet, since it violates the Trusted Path requirement set (see Section 6.1)
- Use of FTP, since it violates the Trusted Path requirement set (see Section 6.1)
- Use of SNMP, since it violates the Trusted Path requirement set (see Section 6.1)
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set (see Section 6.1)
- Media use (other than during installation of the TOE)
- Use of root account, other than during initial installation and configuration.

2 Conformance Claims

2.1 CC Conformance Claim

The TOE is Common Criteria Version 3.1 Revision 3 (July 2009) Part 2 extended and Part 3 conformant.

2.2 PP Claim

The TOE conforms (exact compliance) to the following Protection Profile:

- Security Requirements for Network Devices, Version 1.1, 08June 2012 [NDPP]
- Security Requirements for Network Devices Errata #3, 3 November 2014, [NDPPerr]

It is understood that "exact compliance", as specified in [NDPPerr], is a subset of strict conformance whereby the ST contains all of the requirements in [NDPP] section 4 and the relevant requirements from Appendix C [NDPP]. There is no iteration of requirements in this ST and no additional requirements (from [CC2] or [CC3]) in the ST. Further, no requirements in [NDPP] section 4 are omitted.

The Security Problem definition in this Security Target is consistent with the security problem definition detailed in [NDPP] Section 2. The threats in this ST are the same as the resulting threats detailed in Table 4 of [NDPP] Annex A. The organizational security policies in this ST are the same as those specified in Table 5 of [NDPP] Annex A and the assumptions in this ST are the same as those in Table 3 of [NDPP] Annex A.

The statement of security objectives in this ST is consistent with the statement of security objectives detailed in [NDPP] Section 3. The Security Objectives for the TOE specified in this ST are the same as those in Table 6 of [NDPP] Annex A and the Security Objectives for the Operational Environment specified in this ST are the same as those in Table 7 of [NDPP] Annex A.

The statement of requirement sin this ST is consistent with the statement of requirements detailed in [NDPP] Section 4. The Security Functional Requirements specified in this ST are the same as those in [NDPP] Section 4.2, with all extended requirements taken from [NDPP] Section 4.2. The Security Assurance Requirements specified in this ST include all those in [NDPP] Section 4.3, with all refinements taken from [NDPP] Section 4.3. In addition to those Security Assurance Requirements specified in [NDPP] this ST includes the ASE requirements necessary to evaluate this Security Target as part of a TOE evaluation.

From the additional requirements specified in [NDPP] Annex C, the (extended) requirement FCS_SSH_EXT.1 Explicit SSH is selected. There are no claims for IPSec, TLS or HTTPS included in this ST, so the extended requirements FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1 and FIA_PSK_EXT.1 detailed in Annex C of [NDPP] and [NDPPerr] are not included in this ST. In addition, as there are no separate parts of the TOE the additional requirementFPT_ITT.1 (also specified in [NDPP] Annex C) is not applicable and is not included in this ST.

No requirements are contained in this ST that are in addition to those specified in [NDPP] & [NDPPerr].

3 Security Problem Definition

The security problem to be addressed by the TOE is described by threats and policies that are common to network devices, as opposed to those that might be targeted at the specific functionality of a specific type of network device, as specified in [NDPP].

This chapter identifies assumptions as A.*assumption*, threats as T.*threat* and policies as P.*policy*.

Note that the assumptions, threats, and policies are the same as those found in [NDPP] such that this TOE serves to address the Security Problem.

3.1 Threats

The following threats are addressed by the TOE, as detailed in table 4 of [NDPP] Annex A.

THREAT	DESCRIPTION
T.ADMIN_ERROR	An authorized administrator may incorrectly install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

Table 4 - Threats Addressed by the TOE

3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The TOE is required to meet the following organizational security policies, as specified in table 5 of [NDPP] Annex A.

POLICY NAME	POLICY DESCRIPTION
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

Table 5 - Organizational Security Policies

3.3 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE, as specified in table 3 of [NDPP] Annex A.

ASSUMPTION	DESCRIPTION
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all admin guidance in a trusted manner.

Table 6 - Assumptions

4 Security Objectives

4.1 Security Objectives for the TOE

The IT Security Objectives for the TOE are detailed below, as specified in table 6 of [NDPP] Annex A.

OBJECTIVE	DESCRIPTION
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

Table 7 – TOE Security Objectives

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are detailed below, as specified in table 7 of [NDPP] Annex A.

OBJECTIVE	DESCRIPTION
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all admin guidance in a trusted manner.

Table 8– Operational Environment Security Objectives

4.3 Security Objectives Rationale

As these objectives for the TOE and operational environment are the same as those specified in [NDPP], the rationales provided in the prose of [NDPP] Section 3 and in the tables in [NDPP] Annex A

are wholly applicable to this security target as the statements of threats, assumptions, OSPs and security objectives provided in this security target are the same as those defined in the [NDPP].

5 Extended Security Requirement Components Definition

This section defines the extended Security Functional Requirements (SFRs) to be met by the TOE as drawn from [NDPP].

5.1 Extended TOE Security Functional Requirement Components

This section specifies the extended SFRs for the TOE.

5.1.1 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1 External Audit Trail Storage requires the TSF to use an external IT entity for audit data storage. It is modeled after FAU_STG.1, and is considered to be part of the FAU_STG family.

Management: FAU_STG_EXT.1

There are no management activities foreseen.

Audit: FAU_STG_EXT.1

There are no auditable events foreseen.

FAU_STG_EXT.1 External Audit Trail Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel
FAU_STG_EXT.1.1	The TSF shall be able to [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol.

5.1.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4 Cryptographic key zeroization requires cryptographic keys and cryptographic critical security parameters to be zeroized. It is modeled after FCS_CKM.4, and is considered to be part of the FCS_CKM family.

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

FCS_CKM_EXT.4Cryptographic Key Zeroization

Hierarchical to:	No other components
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.3 FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_RBG_EXT.1 Extended: Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. It is modeled after FCS_COP.1, but belongs to a new family defined for the FCS Class.

Management: FCS_RBG_EXT.1

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

There are no auditable events foreseen.

FCS_RBG_EXT.1 Extended: Random Bit Generation

Hierarchical to: No other components

Dependencies: None

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST16 Special Publication 800-90 using [selection: Hash_DRBG17 (any), HMAC18_DRBG (any), CTR19_DRBG (AES20), Dual_EC21_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

5.1.4 FCS_SSH_EXT.1 Explicit: SSH

FCS_SSH_EXT.1 Extended: SSH requires that SSH be implemented. It belongs to a new family defined for the FCS Class.

Management: FCS_SSH_EXT.1

There are no management activities foreseen.

Audit: FCS_SSH_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Failure to establish a SSH session, and reason for failure;
b) Establishment/Termination of a SSH session, and non-TOE endpoint of connection (IP address) for both successes and failures.

FCS_SSH_EXT.1 Extended: SSH

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)

FCS_COP.1(2) Cryptographic operation (for cryptographic signature)
FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)
FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication)
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_SSH_EXT.1.1	The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254, and [selection: 5656, 6668, no other RFCs].
FCS_SSH_EXT.1.2	The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.
FCS_SSH_EXT.1.3	The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.
FCS_SSH_EXT.1.4	The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms].
FCS_SSH_EXT.1.5	The TSF shall ensure that the SSH transport implementation uses [selection: SSH_RSA, ecdsa-sh2-nistp256] and [selection: PGP-SIGN-RSA, PGP-SIGN-DSS, ecdsa-sha2-nistp384, no other public key algorithms,] as its public key algorithm(s).
FCS_SSH_EXT.1.6	The TSF shall ensure that data integrity algorithms used in SSH transport connection is [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512].
FCS_SSH_EXT.1.7	The TSF shall ensure that diffie-hellman-group14-sha1 and [selection: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

5.1.5 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1 Password Management defines the password strength requirements that the TSF will enforce. It belongs to a new family defined for FIA class.

Management: FIA_PMG_EXT.1

There are no management activities foreseen.

Audit: FIA_PMG_EXT.1

There are no auditable events foreseen.

FIA_PMG_EXT.1 Password Management

Hierarchical to:	No other components
Dependencies:	None
FIA_PMG_EXT.1.1	The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: *other characters*]];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

5.1.6 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is considered to be part of the FIA_UAU family.

Management: FIA_UAU_EXT.2

There are no management activities foreseen.

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) All use of the authentication mechanisms.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

Hierarchical to:	No other components
Dependencies:	None
FIA_UAU_EXT.2.1	The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

5.1.7 FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism, requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is based on a combination of FIA_UAU.1 and FIA_UID.1, and belongs to a new family defined for class FIA.

Management: FIA_UIA_EXT.1

There are no management activities foreseen.

Audit: FIA_UIA_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) All use of the authentication mechanism with provided user identity and origin of the attempt (e.g. IP address).

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

Hierarchical to:	FIA_UID.1 Timing of identification FIA_UAU.1 Timing of Authentication

Dependencies:	None
FIA_UIA_EXT.1.1	The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: ○ Display the warning banner in accordance with FTA_TAB.1; ○ [selection: no other actions, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*]]
FIA_UIA_EXT.1.2	The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.1.8 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords requires administrator passwords to be stored in non-plaintext form and requires the TOE to prevent reading of plaintext passwords. It is modeled after FPT_SSP.2, but it belongs to a new family defined for the FPT class.

Management: FPT_APW_EXT.1

There are no management activities foreseen.

Audit: FPT_APW_EXT.1

There are no audit activities foreseen.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

Hierarchical to:	No other components
Dependencies:	None
FPT_APW_EXT.1.1	The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2	The TSF shall prevent the reading of plaintext passwords.

5.1.9 FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys) requires the TOE to prevent reading of all pre-shared, symmetric, and private keys. It is modeled after FPT_SSP.1, but it belongs to a new family defined for the FPT class.

Management: FPT_SKP_EXT.1

There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

There are no audit activities foreseen.

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

Hierarchical to:	No other components
Dependencies:	None
FPT_SKP_EXT.1.1	The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.10 FPT_TST_EXT.1 Extended: TSF testing

FPT_TST_EXT.1 Extended: TSF testing requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF. It is modeled after FPT_TST.1, but belongs to a new family defined for class FPT.

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no audit activities foreseen.

FPT_TST_EXT.1 TSF testing

Hierarchical to:	No other components
Dependencies:	None
FPT_TST_EXT.1.1	The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

5.1.11 FPT_TUD_EXT.1 Extended: Management of TSF Data

FPT_TUD_EXT.1 Extended: Management of TSF Data, requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation. It belongs to a new family defined for the FPT class.

Management: FPT_TUD_EXT.1

There are no management activities foreseen.

Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Initiation of update.

FPT_TUD_EXT.1 Extended: Trusted Update

Hierarchical to:	No other components
Dependencies:	FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)
FPT_TUD_EXT.1.1	The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

5.1.12 FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking requires system initiated locking of an interactive session after a specified period of inactivity. It is part of the FTA_SSL family.

Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

Hierarchical to:	No other components
Dependencies:	FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism
FTA_SSL_EXT.1.1	The TSF shall, for local interactive sessions, [selection: • lock the session – disable any activity of the user's data access display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session; • terminate the session] after a Security Administrator-specified time period of inactivity.

5.2 Extended TOE Security Assurance Requirement Components

There are no extended TOE Security Assurance Requirement Components required by [NDPP].

6 Security Requirements

This section provides security functional and assurance requirements that must be satisfied by the TOE. These requirements consist of components from the CC Part 2 and Part 3, National Information Assurance Partnership (NIAP) interpreted requirements, and explicit requirements defined in [NDPP]. All extended components are taken from [NDPP] and as such are understood to be defined by [NDPP], hence no statement of extended components is required in this security target.

6.1 Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE, organized by CC class as specified in [NDPP].

Table 8 identifies all the SFR's implemented by the TOE.

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
AUDIT	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	External Audit Trail Storage
CRYPTOGRAPHIC SERVICES	FCS_CKM.1	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4	Cryptographic Key Zeroization
	FCS_COP.1(1)	Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2)	Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4)	Cryptographic Operation (for keyed-hash message authentication)
	FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
	FCS_SSH_EXT.1	Explicit SSH Requirements
USER DATA PROTECTION	FDP_RIP.2	Full residual information protection
IDENTIFICATION & AUTHENTICATION	FIA_PMG_EXT.1	Extended: Password Management
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
SECURITY MANAGEMENT	FMT_MTD.1	Management of TSF Data (for general TSF data)
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on Security Roles
PROTECTION OF THE TOE	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
	FPT_APW_EXT.1.1	Extended: Protection of Administrator Passwords
	FPT_STM.1	Reliable Time Stamps
	FPT_TUD_EXT.1	Extended: Trusted Update
	FPT_TST_EXT.1	TSF Testing
TOE ACCESS	FTA_EXT_SSL.1	TSF-initiated session locking
	FTA_SSL.3	TSF-initiated termination
	FTA_SSL.4	User-initiated termination
	FTA_TAB.1	Default TOE access banners
TRUSTED PATH/CHANNEL	FTP_ITC.1	Inter-TSF trusted channel
	FTP_TRP.1	Trusted path

Table 9– TOE Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit;

c) All administrative actions; and

d) [specifically defined auditable events listed in Table 16, Section 8].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event time, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of ~~Table 1~~Table 16, Section 8].

6.1.1.2 User identity association – human users (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 Protected audit trail storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1 The TSF shall be able to [transmit the generated audit data to an external IT entity] using a trusted channel implementing the [SSH] protocol.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1.1)

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with;

[NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes,

NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-3, "Digital Signature Standard")]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

6.1.2.2 Cryptographic Key Zeroization (for asymmetric keys) (FCS_CKM_EXT.4)

FCS_CKM_EXT.4 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

6.1.2.3 Cryptographic Operation(for data encryption/decryption) (FCS_COP.1)

FCS_COP.1.1(1) The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES operating in [CBC mode]] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, “Advanced Encryption Standard (AES)”
- [NIST SP 800-38A, NIST SP 800-38D]

6.1.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1.1(2) The TSF shall perform cryptographic signature services in accordance with a [:

(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]

that meets the following:

Case: Elliptic Curve Digital Signature Algorithm

- FIPS PUB 186-3, "Digital Signature Standard "
- The TSF shall implement “NIST curves” P-256, P-384 and [P-521] (as defined in FIPS PUB 186-3, “Digital Signature Standard”).]

Application Note: ECDSA (P-256) + SHA256 is used for package verification by EX/QFX-series, as required for FPT_TUD_EXT.1. ECDSA signature services are also used by the SSH module, in support of the FCS_SSH_EXT.1 requirements.

6.1.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1.1(3) The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-512] and message digest sizes [160, 256, 512] bits that meet the following: FIPS Pub 180-3, “Secure Hash Standard.”

6.1.2.6 Cryptographic Operation (for key-hash message authentication) (FCS_COP.1(4))

FCS_COP.1.1(4) The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm HMAC-[SHA1, SHA-256, SHA-512], key size [*160, 256, 512 bits*], and message digest sizes [160, 256, 512] bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, “Secure Hash Standard.”

6.1.2.7 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [NIST Special Publication 800-90 using [HMAC_DRBG (any)]] seeded by an

entropy source that accumulated entropy from [a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

6.1.2.8 Explicit: SSH (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254, and [5656, 6668].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256K*] bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [no other algorithms].

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [ecdsa-sha2-nistp256] and [no other public key algorithms] as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [hmac-sha1, hmac-sha2-256, hmac-sha2-512].

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 and [ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521] are the only allowed key exchange methods used for the SSH protocol.

6.1.3 User Data Protection (FDP)

6.1.3.1 Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [allocation of the resource to] all objects.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”];

2. Minimum password length shall **be** settable by the **Authorized**[6] Administrator, and support passwords of 15 characters or greater;

6.1.4.2 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [[*routing/switching services, including ping, arp, BFD send (UDP port 49152), GRE OAM Keep-alive and SGR tunnel status (UDP port 49153) and HCM JVAS plugin (UDP port 49154) services*]].

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

6.1.4.3 Extended: Password-based Authentication mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [public key-based authentication] to perform administrative user authentication.

Application Note: ECDSA is the public key algorithm supported for administrative user authentication.

6.1.4.4 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress **at the local console**[7].

6.1.5 Security Management (FMT)

6.1.5.1 Management of TSF data (For General TSF data) (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to the **Authorized**[6] Administrators.

6.1.5.2 Specification of management functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;

[6]This is identified as a refinement as the PP uses the term "Security Administrator" in this instance, but defines the role "Authorized Administrator" in FMT_SMR.1 (see section 6.1.5.3). Therefore, the ST has adopted and applied the term "Authorized Administrator" for consistency reasons.
[7]The refinement "at the local console" is not marked in [NDPP].

- [No other capabilities].

Application Note: ECDSA is the supported digital signature algorithms (as specified in FCS_COP.1(2)) for NDPP compliance.

6.1.5.3 Restrictions on security roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Authorized Administrator.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Extended: Protection of TSF Data (for reading of all symmetric keys)(FPT_SKP_EXT.1)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

6.1.6.2 Extended: Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

6.1.6.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

6.1.6.4 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide **authorized**[6] administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide **authorized**[6] administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

6.1.6.5 Extended: TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

6.1.7 TOE Access (FTA)

6.1.7.1 TSF-initiated session locking (local sessions) (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [terminate the session] after **an Authorized**[6] Administrator-specified time period of inactivity.

6.1.7.2 TSF-initiated termination (remote sessions) (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate a remote interactive session after **an [Authorized**[6] Administrator-configurable time interval of session inactivity].

6.1.7.3 User-initiated termination (FTA_SSL_EXT.4)

FTA_SSL.4.1 The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

6.1.7.4 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing an administrative user session the TSF shall display **an Authorized**[6] Administrator-specified advisory notice and consent warning message regarding use of the TOE.

6.1.8 Trusted Path/Channels (FTP)

6.1.8.1 Inter-TSF trusted channel (prevention of disclosure) (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall use [SSH] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [[*no other capabilities*]] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2 The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*export of audit logs to syslog servers*].

6.1.8.2 Trusted path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall use [SSH] to provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2 The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

6.2 Security Assurance Requirements

This section defines the assurance requirements for the TOE, which are summarized in Table 10 below.

The security assurance requirements included in this Security Target include all those specified in [NDPP] for which conformance is claimed. In addition, Table 10 details the ASE Security Assurance Requirements to be applied for the evaluation of this ST, in the context of a TOE evaluation.

ASSURANCE CLASS	COMPONENTS	DESCRIPTION
ASE: Security Target	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.2	Security objectives
	ASE_ECD.1	Extended components definition
	ASE_REQ.2	Derived security requirements
	ASE_TSS.1	TOE Summary Specification
ADV: Development	ADV_FSP.1	Basic functional specification
AGD: Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
ALC: Lifecycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
ATE: Tests	ATE_IND.1	Independent Testing – Conformance
AVA: Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 10 – Security Assurance Requirements

6.3 Security Requirements Rationale

6.3.1 Security Functional Requirements Rationale

The rationale of how the security functional requirements meet all objectives for the TOE is provided in the prose of [NDPP] Section 3. As all objectives and all SFRs in this Security Target are the same as those specified in [NDPP] the rationale provided in [NDPP] Section 3 is wholly applicable to this security target.

All dependencies of security functional requirements are satisfied as demonstrated in below.

SFR	Dependency	Satisfaction of dependency
FAU_GEN.1	FPT_STM.1 Reliable time stamps	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	FAU_GEN.1 FIA_UID.1 dependency satisfied by FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FAU_STG_EXT.1	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel	FAU_GEN.1 FTP_ITC.1

SFR	Dependency	Satisfaction of dependency
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1 (1-4) FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_CKM_EXT.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1
FCS_COP.1(1)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_COP.1(2)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_COP.1(3)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_COP.1(4)	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4 dependency met by FCS_CKM_EXT.4
FCS_RBG_EXT.1	None	n/a
FCS_SSH_EXT.1	FCS_TLS_EXT.1 Extended: TLS	FCS_TLS_EXT.1

SFR	Dependency	Satisfaction of dependency
FDP_RIP.2	FCS_COP.1(1) Cryptographic operation (for data encryption/decryption) FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing) FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication) FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation) FCS_CKM.1 Cryptographic Key Generation FCS_CKM_EXT.4 Cryptographic Key Zeroization	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3)) FCS_COP.1(4) FCS_RBG_EXT.1 FCS_CKM.1 FCS_CKM_EXT.4
FIA_PMG_EXT.1	FCS_COP.1(1) Cryptographic operation (for data encryption/decryption) FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing) FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication) FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation) FCS_CKM.1 Cryptographic Key Generation FCS_CKM_EXT.4 Cryptographic Key Zeroization	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3)) FCS_COP.1(4) FCS_RBG_EXT.1 FCS_CKM.1 FCS_CKM_EXT.4
FIA_UIA_EXT.1	None	n/a
FIA_UAU_EXT.2	None	n/a
FIA_UAU.7	None	n/a
FMT_MTD.1	None	n/a
FMT_SMF.1	FIA_UAU.1 Timing of authentication	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FMT_SMR.2	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FPT_SKP_EXT.1	None	n/a

SFR	Dependency	Satisfaction of dependency
FPT_APW_EXT.1.1	FIA_UID.1 Timing of identification	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FPT_STM.1	None	n/a
FPT_TUD_EXT.1	None	n/a
FPT_TST_EXT.1	None	n/a
FTA_SSL_EXT.1	None	n/a
FTA_SSL.3	FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)	FCS_COP.1(2) FCS_COP.1(3)
FTA_SSL.4	None	n/a
FTA_TAB.1	FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism	FIA_UIA_EXT.1
FTP_ITC.1	None	n/a
FTP_TRP.1	None	n/a

Table 11– Satisfaction of dependencies

6.3.2 Security Assurance Requirements Rationale

The rationale provided in [NDPP] Section 4.3 for the selection of security assurance requirements is wholly applicable to this security target, as the security assurance requirements specified in this security target are the same as those specified in [NDPP].

7 TOE Summary Specification

This section provides summary information on how the security requirements are met. The objective is to give a high-level view of the security requirements are satisfied by the TOE; therefore, the descriptions are not overly detailed.

7.1 Security Audit

Junos OS creates and stores audit records for the following events (the detail of content recorded for each audit event is detailed in Table 16):

a) Start-up and shutdown of the audit function[8];
b) All administrative actions;
c) All events specified in Table 16.

Auditing is done using syslog. Syslog can be configured to store the audit logs locally, and optionally to send them to one or more syslog log servers (via Netconf over SSH[9]). Local audit log are stored in /var/log/ in the underlying filesystem. Only an authorized administrator can read log files, or delete log and archive files through the CLI interface or through direct access to the filesystem having first authenticated as an authorized administrator (see Section 7.4 below). The syslogs are automatically deleted locally according to configurable limits on storage volume.

The TOE defines an active log file and a number of "archive" files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file 'logfile.0.gz'. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size, 'logfile.0.gz' is renamed 'logfile.1.gz', and the active log file is closed, compressed, and renamed 'logfile.0.gz' (see [SG_EX] Chapter 23 Subsection "file"). When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.

For EX-series switches the default maximum size is 128KB. The default maximum size can be modified by the user, as detailed in [SG_EX] Chapter 23 Subsection "size (System)".

A 1MB syslog file takes approximately 0.25Mb of storage when archived. Syslog files can acquire complete storage allocated to /var filesystem which is platform specific. However, when the filesystem reaches 92% storage capacity an event is raised to the administrator but the eventd process (being a privileged process) still can continue using the reserved storage blocks. This allows the syslog to continue storing events while the administrator frees the storage. If the administrator does not free the storage in time and the /var filesystem storage becomes exhausted a final entry is recorded in the log reporting "No space left on device" and logging is terminated. The appliance continues to operate in the event of exhaustion of audit log storage space. The QFX5100 and EX4600 allocate at least 484Mb for the storage of audit files.

For more information about configuring event logging see [SG_EX] and [ECG14.1].

[8] Start-up and shutdown of the audit function are synonymous with start-up and shutdown of the TOE, as the audit functions cannot be enabled or disabled, and so form part of the TOE start-up and shutdown process, respectively.
[9] In accordance with RFC 4741.

The Audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1
- FAU_GEN.2
- FAU_STG_EXT.1

7.2 Cryptographic Support

All FIPS-approved cryptographic functions implemented by the secure network appliance are implemented in the Junos crypto module. The TOE evaluation provides a CAVP validation certificate for all FIPS-approved cryptographic functions implemented by the TOE. CAVP certificate details are provided in Table 12 – CAVP Certificate Results, below

Implementation	Algorithm	Cert Number
MD (libMD)	SHA	*#3070*
	HMAC	*#2402*
OpenSSL	AES	*#3654*
	DSA	*#1025*
	ECDSA	*#762*
	SHA	*#3071*
	HMAC	*#2403*
	DRBG-HMAC	*#983*

Table 12 – CAVP Certificate Results

The TOE meets the cryptographic requirements by allowing the administrator to run a FIPS install package (per platform guidance). The evaluated configuration of the TOE details that the FIPS operating mode should not be enabled[10]. The Cryptographic security function is described in the context of how it satisfies the cryptographic security requirements.

The crypto module implements Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater (as specified by the authorized administrator) with SHA256 for digital signature generation and verification.

The TOE implements a timeout period for authentication for the SSHv2 protocol and provides a limit of three failed authentication attempts. The TOE uses public key-based authentication methods and password-based authentication for SSHv2.

Packets greater than 256Kbytes in an SSH transport connection are dropped and the connection is terminated by the TOE.

The TOE supports AES-CBC-128 and AES-CBC-256 encryption algorithms for SSH transport and uses “ecdsa-sha2-nistp256” as its public key algorithm.

[10] The knob “set system fips level 1” (which is NOT set in the evaluated configuration) will enforce strict compliance to FIPS and enable restrictions on algorithms and keys sizes as required by FIPS requirements. While FIPS validated algorithms are invoked to provide the cryptographic operations necessary to support the evaluation configuration (including encryption, decryption, hashing services, signature services, random number generation and self-testing), FIPS mode should not be applied.

The data integrity algorithms used in SSH transport connection are"hmac-sha1"as required by [RFC4253] and hmac-sha2-256 and hmac-sha2-512 as required by [RFC6668].

Key exchange is done using one of "diffie-hellman-group14-sha1" [RFC4253] ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521 [RFC5656]. No other key exchange methods are supported in the evaluated configuration, as detailed in [ECG14.1].

The TOE supports cryptographic hashing via the SHA-1, SHA-256 SHA-512 algorithms, provided it has a message digest size of either 160, 256 or 512 bits.

The TOE handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 13– Key zeroization handling below. Zeroization is performed when then memory is called back for subsequent use, and is zeroized before it is re-used.

Junos OS performs random number generation in accordance with NIST Special Publication 800-90 using HMAC_DRBG, SHA-256. This includes input of timing information from process execution time stored by CentOS in an entropy pool for the host OS (Junos OS VM) which is used by the Junos OS VM as an additional source of entropy.

CSP	Description	How Stored	Where Stored	Zeroization Method
SSH Private Host Key	The first time SSH is configured, the key is generated. Used to identify the host.	Plaintext	Disk	When the appliance is recommissioned, the config files (including CSP files such as SSH keys) are removed using the zeroise option. Files are overwritten three times using the zeroize option, before they are deleted
SSH Private Host Key	Loaded into memory to complete session establishment	Plaintext	Memory	Memory is overwritten upon session termination (when released by the Junos VM, the Qemu hypervisor erases the released memory before it is placed in the free pool)
SSH Session Key	Session keys used with SSH, AES 128, 256, HMAC-SHA-1, hmac-sha2-256 or hmac-sha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	Memory is overwritten upon session termination (when released by the Junos VM, the Qemu hypervisor erases the released memory before it is placed in the free pool)
User Password	Plaintext value as entered by user	Plaintext as entered Hashed when stored	Processed in Memory Stored on disk	When released by the Junos VM, the Qemu hypervisor erases the released memory before it is placed in the free pool. When the appliance is recommissioned, the config files (including the obfuscated password) are removed using the zeroise option.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's
ecdh private keys	Loaded into memory to complete key exchange in session establishment	Plaintext	Memory	Memory is overwritten upon session termination (when released by the Junos VM, the Qemu hypervisor erases the released memory)

Table 13– Key zeroization handling

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1
- FCS_CKM_EXT.4
- FCS_COP.1(1)

- FCS_COP.1(2)
- FCS_COP.1(3)
- FCS_COP.1(4)
- FCS_RBG_EXT.1
- FCS_SSH_EXT.1

7.3 User Data Protection

The only resource made available to information flowing through a TOE is the temporary storage of packet information when access is requested and when information is being routed. User data is not persistent when resources are released by one user/process and allocated to another user/process. Temporary storage (memory) used to build network packets is erased when the resource is called into use by the next user/process. Junos knows, and keeps track of, the length of the packet. This means that when memory allocated from a previous user/process arrives to build the next network packet, Junos is aware of when the end of the packet is reached and pads a short packet with zeros accordingly. Hence, the memory content is overwritten by either the content of the subsequent packet or zeros and no residual information from packets in a previous information stream can traverse through the TOE.

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2

7.4 Identification and Authentication

The TSF enforces binding between human users and subjects. The Authorized Administrator is responsible for provisioning user accounts, and only the Authorized Administrator can do so. User accounts in the TOE have the following attributes: user identity (user name), authentication data (password) and role (privilege). The Authorized Administrator is associated with a defined login class, which is assigned “permissions all”.

Junos users are configured under “system login user” and are exported to the password database ‘/var/etc/master.passwd’. A Junos user is therefore an entry in the password database. Each entry in the password database has fields corresponding to the attributes of “system login user”, including username, (obfuscated) password and login class. The passwords are stored in obfuscated form using sha1 or sha-256, as detailed in [ECG14.1].

Locally stored authentication data for fixed password authentication is a case-sensitive, alphanumeric value. The password has a minimum length of 15[11] characters, must contain characters from at least two different character sets (upper, lower, numeric, punctuation), and can be up to 20 ASCII characters in length (control characters are not recommended).Authentication data for public key-based authentication methods are stored in a directory owned by the user (and typically with the same name as the user). This directory contains the files ‘.ssh/authorized_keys’ and ‘.ssh/authorized_keys2’ which are used for SSH public key authentication.

The internal architecture supporting Authentication includes an active process, associated linked libraries and supporting configuration data. The Authentication process and library are

[11] By default the minimum password length is 10, but this is configurable and can be set to another minimum length value, e.g. 15 using the command: set system login password minimum-length 15

- login()
- PAM Library module

Following TOE initialization, a 'login' process is listening for a connection at the local console. This 'login' process can be accessed through either direct connection to the local console or following successful establishment of a remote management connection over SSH (as detailed in Section 7.8), when a login prompt is displayed.

This login process identifies and authenticates the user using PAM operations. The login process does two things; it first establishes that the requesting user is whom they claim to be and second provides them with an interactive Junos Command interactive command line interface (CLI).

The SSH daemon supports public key authentication by looking up a public key in an authorized keys file located in the directory '.ssh' in the user's home directory (i.e. '~/.ssh/') and this authentication method will be attempted before any other if the client has a key available. The SSH daemon will ignore the authorized keys file if it or the directory '.ssh' or the user's home directory are not owned by the user or are writeable by anyone else.

For password authentication, login() interacts with a user to request a username and password to establish and verify the user's identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed. Login() uses PAM Library calls for the actual verification of this data.The password is hashed and compared to the stored value, and success/failure is indicated to login(). PAM is used in the TOE support authentication management, account management, session management and password management. Login primarily uses the session management and password management functionality offered by PAM.

Following authentication, login launches the CLI using an exec()[12] system call. Such an invocation, results in the main() function for the CLI to be invoked.

The TOE requires users to provide unique identification and authentication data (passwords/public key) before any access to the system is granted. A password is configured for each user allowed to log into the secure switch. The TOE successfully authenticates if the authentication data provided matches that stored in conjunction with the provided identity.

The TOE will permit support of the following services prior to identification and authentication of the administrator: ping, arp, BFD send (UDP port 49152), GRE OAM Keep-alive and SGR tunnel status (UDP port 49153) and HCM JVAS plug-in (UDP port 49154). These services are permitted by default once the evaluated configuration, as specified in [ECG14.1], has been applied. No administrator functions are available prior to identification and authentication. Junos OS process permissions prevent the daemons associated with these listening services from accessing any TSF data.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.2
- FIA_UAU.7

[12] Any of the exec family of system calls may be used.

7.5 Security Management

There is only one user role defined for the TOE: Authorized Administrator. The Authorized Administrator is responsible for provisioning user accounts. User accounts in the TOE have the following attributes: user identity (user name), authentication data (password/public key) and role (privilege). Locally stored authentication data for fixed password authentication is a case-sensitive, value comprised of any combination of upper and lower case letters, numbers, and punctuation (from the set [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”). Public keys are stored in ‘.ssh’ files in the user’s home directory (i.e. ‘~/.ssh/’).

The TOE provides user access either through the system console or remotely over the Trusted Path using the SSHv2 protocol. Users are required to provide unique identification and authentication data (passwords/public key) before any access to the system is granted. A password is configured for each user allowed to log into the secure switch. Password information is stored as hashed data (using hmac-sha1) in the authentication database and public keys are stored in plaintext in ‘.ssh’ files in the user’s home directory (i.e. ‘~/.ssh/’). The TOE successfully authenticates if the authentication data provided matches that stored in conjunction with the provided identity.

The Authorized Administrator has the capability to:

- Modify cryptographic security data (import of certificates for the establishment of SSH sessions) and date/time
- Restrict the service available to unidentified or unauthenticated IT entities
- Restrict TOE (release) updates[13]

Detailed topics on the secure management of Juniper EX-series switches are discussed in [SG_EX] and [ECG14.1].

The Security Management function is designed to satisfy the following security functional requirements:

- FMT_MTD.1
- FMT_SMF.1
- FMT_SMR.2

7.6 Protection of the TSF

The clock function of the TOE provides a source of date and time information for the appliance, used in audit timestamps. CentOS kernel provides the current time when it bootstraps the Junos OS VM. Once the Junos OS VM is started it maintains its own time using the hardware Time Stamp Counter as the clock source.

For each user session the TOE maintains a count of clock cycles (provided by the system clock) since last activity. The count is reset each time there is activity related to the user session. When the counter reaches the number of clock cycles equating to the configured period of inactivity the user session is locked out.

[13]Patch updates are not included in the scope of the evaluation; only complete release updates are supported.

Authorized administrators are able to query the current version of the TOE firmware/software. Junos does not provide partial updates for the TOE, customers requiring updates must migrate to a subsequent release.

The install package includes both the Junos OS and the CentOS virtualization kernel. These cannot be updated separately in the evaluated configuration; they must be installed as a single package. Once the Junos OS VM is loaded the procedures detailed in [ECG14.1] will be applied to disable the loading of additional VMs.

The Junos OS kernel maintains a set of fingerprints (SHA1 digests) for executable files and other files which should be immutable. No executable can be run or shared object loaded unless the fingerprint is correct. The fingerprints are loaded as the filesystems are mounted, from digitally signed manifests. The manifest file is signed using the Juniper engineering private key, and is verified by the TOE using the Juniper engineering public key (stored on the TOE filesystem in clear, protected by filesystem access rights). ECDSA (P-256) with SHA-256 is used for package verification.

The fingerprint loader will only process a manifest for which it can verify the signature. Thus without a valid digital signature an executable cannot be run. When the command is issued to install an update (e.g. request system software add jinstall), the manifest file for the update is verified and stored, and each executable/immutable file is verified before it is executed. If any of the fingerprints in an update are not correctly verified, the TOE rolls back to the last known verified image.

Integrity checking of firmware includes an integrity check to verify the integrity of the CentOS kernel image and the CentOS root filesystem.

The Junos OS VM will run the following set of self-tests during power on to check the correct operation of the Junos OS portions of the TOE:

- Power on test – determines the boot-device responds, and performs a memory size check to confirm the amount of available memory.
- File integrity test –verifies integrity of all mounted signed packages, to assert that system files have not been tampered with.
- Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys and iked credentials, such as Cas, CERTS, and various keys.
- Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.
- Kernel, libmd, OpenSSL, QuickSec, SSH Ipsec – verifies correct output from known answer tests for appropriate algorithms

The power on self-tests are run in different modules for example:

Testing **kernel KATS**:
DES3-CBC Known Answer Test
HMAC-SHA1 Known Answer Test
HMAC-SHA2-256 Known Answer Test
SHA-2 Known Answer Test
AES128-CMAC Known Answer Test
AES-CBC Known Answer Test

Testing **MacSec KATS**:
AES128-CMAC Known Answer Test

Testing **libmd KATS**:
HMAC-SHA1 Known Answer Test
HMAC-SHA2-256 Known Answer Test
SHA-2 Known Answer Test

Testing **OpenSSL KATS**:
FIPS RNG Known Answer Test
NIST 800-90 HMAC DRBG Known Answer Test
FIPS DSA Known Answer Test
FIPS ECDSA Known Answer Test
FIPS ECDH Known Answer Test
FIPS RSA Known Answer Test
DES3-CBC Known Answer Test
HMAC-SHA1 Known Answer Test
HMAC-SHA2-224 Known Answer Test
HMAC-SHA2-256 Known Answer Test
HMAC-SHA2-384 Known Answer Test
HMAC-SHA2-512 Known Answer Test
SHA-2 Known Answer Test
AES-CBC Known Answer Test
AES-GCM Known Answer Test
ECDSA-SIGN Known Answer Test
KDF-IKE-V1 Known Answer Test
KDF-SSH Known Answer Test

Testing **QuickSec KATS**:
NIST 800-90 HMAC DRBG Known Answer Test
DES3-CBC Known Answer Test
HMAC-SHA1 Known Answer Test
HMAC-SHA2-224 Known Answer Test
HMAC-SHA2-256 Known Answer Test
HMAC-SHA2-384 Known Answer Test
HMAC-SHA2-512 Known Answer Test
AES-CBC Known Answer Test
SSH-RSA-ENC Known Answer Test
SSH-RSA-SIGN Known Answer Test
KDF-IKE-V1 Known Answer Test
KDF-IKE-V2 Known Answer Test

Testing **SSH Ipsec KATS**:
NIST 800-90 HMAC DRBG Known Answer Test
DES3-CBC Known Answer Test
HMAC-SHA1 Known Answer Test
HMAC-SHA2-256 Known Answer Test
SHA-2 Known Answer Test
AES-CBC Known Answer Test
SSH-RSA-ENC Known Answer Test
SSH-RSA-SIGN Known Answer Test
KDF-IKE-V1 Known Answer Test

Testing **file integrity**:
File integrity Known Answer Test

Testing **crypto integrity**:
Crypto integrity Known Answer Test

Expect an exec Authentication error…
/sbin/kats/run-tests: /sbin/kats/cannot-exec: Authentication error

Junos OS is designed to fail securely. In the event of a transiently corrupt state or failure condition, the system will report an error; the event will be logged and the system restarted, having ceased to process network traffic. When the system restarts, the system boot process does not succeed without passing all self-tests for cryptographic algorithms, RNG tests, and software integrity tests. The logging of this self-test behavior is discussed in Chapter 10 of [ECG14.1].

The TOE does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission[14].

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_STM.1
- FPT_TUD_(EXT).1
- FPT_TST_EXT.1

7.7 TOE Access

Junos enables Authorized Administrators to configure an access banner provided with the authentication prompt. The banner can provide warnings against unauthorized access to the secure switch as well as any other information that the Authorized Administrator wishes to communicate.

User sessions can be terminated by users. The Authorized Administrator can set the TOE so that a user session is terminated after a period of inactivity.

The TSF overwrites the display device and makes the current contents unreadable after the local interactive session is terminated due to inactivity, thus disabling any further interaction with the TOE. This mechanism is the inactivity timer for administrative sessions. The Authorized Administrator can configure this inactivity timer on administrative sessions after which the session will be logged out.

The local administrative user can logout of existing session by typing logout to exit the CLI admin session and the TSF makes the current contents unreadable after the admin initiates the termination. No user activity can take place until the user re-identifies and authenticates.

The TOE Access function is designed to satisfy the following security functional requirements:

- FTA_SSL_EXT.1.1
- FTA_SSL.3
- FTA_SSL.4
- FTA_TAB.1

[14] [ECG14.1] details the use of the root user is limited to initial installation and configuration and is not to be used in normal operation.

7.8 Trusted Path/Channels

The TOE supports and enforces Trusted Channels that protect the communications between the TOE and a remote audit server from unauthorized disclosure or modification. It also supports Trusted Paths between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification.

The TOE achieves Trusted Channels by use of the SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. Either the TOE or the remote audit server can initiate the connection, and mutual identification of the endpoints is guaranteed by using public key certificate based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

The TOE achieves Trusted Paths by use of the SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between the TSF and a remote administrator is provided by the use of an SSH session. Remote administrators of the TSF initiate communication with the TSF through the SSH tunnel created by the SSH session. Assured identification of the TSF is guaranteed by using public key based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

Local console access is gained by connecting an RJ-45 cable between the console port on the appliance and a workstation with a serial connection client.

The Trusted Path/Channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1
- FPT_TRP.1

7.9 RFC Conformance Statements

This section identifies, for the critical RFCs applied in the implementation of SSH, the options supported by the TOE.

RFC	RFC synopsis	TOE Handling of Security-Related Protocol Options
RFC 4251	The Secure Shell (SSH) Protocol Architecture	**Host Keys:** The TOE uses an ECDSA Host Key for SSH v2, which is generated on initial setup of the TOE. Any of them can be de-configured via the CLI and the key will be deleted and thus unavailable during connection establishment. This key is randomly generated to be unique to each TOE instance. The TOE presents the client with its public key and the client matches this key against its known_hosts list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different (per the SSHv2 protocol). **Policy Issues:** The TOE implements all mandatory algorithms and methods. The TOE can be configured to accept public-key based authentication and/or password-based authentication. The TOE does not require multiple authentication mechanisms for users. The TOE allows port forwarding and sessions to clients. The TOE has no X11 libraries or applications and X11 forwarding is prohibited. **Confidentiality:** The TOE does not accept the “none” cipher. For ciphers whose blocksize >= 16, the TOE rekeys every 2^32 blocks have been sent/received. For other ciphers, the TOE rekeys connections, after 2^27 blocks have been sent/received. (Rekeying can also be triggered by sending 2^31 + 1 packets, rather than blocks.) The client may explicitly request a rekeying event as a valid SSHv2message at any time and the TOE will honor this request. **Denial of Service:** When the SSH connection is brought down, the TOE does not attempt to re-establish it. **Ordering of Key Exchange Methods:** The TOE orders key exchange algorithms as follows: ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521, diffie-hellman-group14-sha1. **Debug Messages:** The TOE sshd server does not support debug messages via the CLI. **End Point Security:** The TOE permits port forwarding. **Proxy Forwarding:** The TOE permits proxy forwarding. **X11 Forwarding:** The TOE does not support X11 forwarding.

RFC	RFC synopsis	TOE Handling of Security-Related Protocol Options
RFC 4252	The Secure Shell (SSH) Authentication Protocol	**Authentication Protocol:** The TOE does not accept the "none" authentication method. The TOE disconnects a client after 30 seconds if authentication has not been completed. The TOE also allows authentication retries of three times before sending a disconnect to the client. **Authentication Requests:** The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed – it sends back a disconnect as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies "none" authentication method and replies with a list of permitted authentication methods. **Public Key Authentication Method:** The TOE supports public key authentication. Authentication succeeds if the correct private key is used. The TOE does not require multiple authentications (public key and password) for users. **Password Authentication Method:** The TOE supports password authentication. Expired passwords are not supported and cannot be used for authentication. **Host-Based Authentication:** The TOE does not support the configuration of host-based authentication methods.
RFC 4253	The Secure Shell (SSH) Transport Layer Protocol	**Encryption:** The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc[15]. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the "none" algorithm for encryption. **Data Integrity:** The TOE permits negotiation of HMAC-SHA1in each direction. **Key Re-Exchange:** The TOE performs a re-exchange when SSH_MSG_KEXINIT is received.
RFC 4254	Secure Shell (SSH) Connection Protocol	**Multiple channels:** The TOE assigns each channel a number (as detailed in RFC 4251, see above). **Data transfers:** The TOE supports a maximum window size of 256K bytes for data transfer. **Interactive sessions:** The TOE only supports interactive sessions that do NOT involve X11 forwarding. **Forwarded X11 connections:** This is not supported in the TOE. **Environment variable passing:** The TOE only sets variables once the server process has dropped privileges. **Starting shells/commands:** The TOE supports starting one of shell, application program or command (only one request per channel). These will be run in the context of a channel, and will not halt the execution of the protocol stack. **Window dimension change notices:** The TOE will accept notifications of changes to the terminal size (dimensions) from the client. **Port forwarding:** This is fully supported by the TOE.

[15]Others are supported by default, but these are the encryption algorithms [ECG14.1] specifies are to be configured in the evaluated configuration.

RFC	RFC synopsis	TOE Handling of Security-Related Protocol Options
RFC5656	SSH ECC Algorithm Integration	**ECDH Key Exchange:** The client matches the key against its known_hosts list of keys. **Required Curves:** All required curves are implemented: ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521. None of the recommended curves are supported as they are not included in [NDPPerr].
RFC 6668	sha2-Transport Layer Protocol	**Data Integrity Algorithms:** Both the recommended and optional algorithms hmac-sha2-256 and hmac-sha2-512 (respectively) are implemented.

Table 14 – RFC Conformance Statements

The RFC conformance statements support the satisfaction of FCS_SSH_EXT.1.

7.10 Conformance Statements for 800-56

The following sections detail all sections of the [800-56A] standard the TOE complies with for generation of asymmetric cryptographic keys (as claimed in FCS_CKM.1). The relevant sections of [800-56A] are section 5.5 “Domain Parameters” and section 5.6 “Private and Public Keys”.

All “SHALL” statements within the listed sections are implemented in the TOE and all “SHALL NOT” statements are adhered to within the TOE and the described functionality/behavior is not present. The implemented option associated with each “SHOULD” and “SHOULD NOT” statement in a referenced section is detailed.

There are no TOE specific extensions relating to cryptographic key generation that are not included in this standard.

7.10.1 Finite Field-Based and Elliptic Curve-Based Key Establishment Schemes

The requirements for both Finite Field-Based Key Establishment Schemes and Elliptic Curve-Based Key Establishment Schemes are specified in [800-56A]:

800-56A section	800-56A sub section	Compliance
5.5 Domain Parameters	General	Comply with all “shall” statements.
5.5.1 Domain Parameter Generation	5.5.1.1 FFC Domain Parameter Generation	Comply with all “shall” statements.
	5.5.1.2 ECC Domain Parameter Generation	Comply with all “shall” statements.
5.6 Key Establishment Key Pairs	General	No statements
5.6.1 Key Pair Generation	5.6.1.1 FFC Key Pair Generation	Comply with all “shall” statements.
	5.6.1.2 ECC Key Pair Generation	Comply with all “shall” statements.

800-56A section	800-56A sub section	Compliance
5.6.2 Required Assurances	General	Comply with all “shall” statements. The TOE will determine and explicitly reflect whether or not key establishment is allowed based upon the method(s) of assurance that was used.
	5.6.2.1 Assurances Required by the Key Pair Owner	Owner Receives Assurance via Key Generation – The act of generating a key pair. Owner Full Validation – The owner performs a successful full public key validation, via pair-wise consistency check If consistency fails the key pair shall not be used.
	5.6.2.2 Assurances Required by a Public Key Recipient	The recipient receives assurance that a trusted third party (trusted by the recipient) has generated the public/private key pair in accordance with Section 5.6.1 and has provided the key pair to the owner. The TOE will be made aware of the method(s) used by the third party. The underlying key agreement used by the TOE is “dhOneFlow or (Cofactor) One-Pass Diffie-Hellman”. Comply with all “shall” statements.
	5.6.2.3 Public Key Validation Routines	Comply with all “shall” statements.

800-56A section	800-56A sub section	Compliance
5.6.3 Key Pair Management	5.6.4.1 Common Requirements on Static and Ephemeral Key Pairs	Comply with all “shall” statements and the “shall not” statement.
	5.6.4.2 Specific Requirements on Static Key Pairs	Comply with all “shall” statements and the “shall not” statement. In item #3 – The TOE will determine whether or not key establishment is allowed based upon the method(s) of assurance that was used.
	5.6.4.3 Specific Requirements on Ephemeral Key Pairs	Comply with all “shall” statements. In item #2 – The TOE will generate an ephemeral key pair just before the ephemeral public key is transmitted. In item #3 – The TOE will determine whether or not to key establishment is allowed based upon the method(s) of assurance that was used.

Table 15 – [800-56A] Conformance Statements

8 Audit Events

The table below maps security requirements to auditable events and audit record contents, in support of FAU_GEN.1.1.

REQUIREMENT	AUDITABLE EVENTS	AUDIT RECORD CONTENTS
FAU_GEN.2	None	
FAU_STG_EXT.1	None	
FCS_CKM.1	None	
FCS_CKM_EXT.4	None	
FCS_COP.1(1)	None	
FCS_COP.1(2)	None	
FCS_COP.1(3)	None	
FCS_COP.1(4)	None.	
FCS_RBG_EXT.1	None.	
FCS_SSH_EXT.1	Failure to establish an SSH session Establishment/Termination of an SSH session	Reason for failure: • Protocol version mismatch • cipher mismatch between client and server • mac algorithm mismatch • ssh hostkey mismatch • ssh key-exchange mismatch Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.

REQUIREMENT	AUDITABLE EVENTS	AUDIT RECORD CONTENTS
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 16 – Security Audit Requirements

9 Install Packages

This section details the install packages for the EX/QFX switches (combining both CentOS kernel and Junos OS VM images).

- EX4600 (Intel Xeon i386 process platform)-> jinstall-ex-4600-junos14.1X53-D30.3-domestic-signed.tgz
- QFX5100 (Intel Xeon i386 process platform)-> jinstall-qfx-5-14.1X53-D30.3-domestic-signed.tgz

In addition the FIPS install package that must be applied to the above switch images, namely:

- fips-mode-i386-14.1X53D30.3-signed.tgz

10 TOE Network Interface Options

10.1 EX4600

Model Number	Description
QFX-SFP-10GE-SR	SFP+ 10GBASE-SR 10 Gigabit Ethernet Optics, 850 nm for up to 300 m transmission on multimode fiber (MMF)
QFX-SFP-10GE-USR	SFP+ 10 Gigabit Ethernet Ultra Short Reach Optics, 850 nm for 10 m on OM1, 20 m on OM2, 100 m on OM3 multimode fiber (MMF)
QFX-SFP-10GE-LR	SFP+ 10GBASE-LR 10 Gigabit Ethernet Optics, 1,310 nm for 10 km transmission on single mode fiber-optic (SMF)
QFX-SFP-10GE-ER	SFP+ 10GBASE-ER 10 Gigabit Ethernet Optics, 1,550 nm for 40 km transmission on single-mode fiber (SMF)

10.2 QFX5100

Model Number	Description
QFX-SFP-10GE-SR	SFP+ 10GBASE-SR 10 Gigabit Ethernet Optics, 850 nm for up to 300 m transmission on multimode fiber (MMF)
QFX-SFP-10GE-USR	SFP+ 10 Gigabit Ethernet Ultra Short Reach Optics, 850 nm for 10 m on OM1, 20 m on OM2, 100 m on OM3 multimode fiber (MMF)
QFX-SFP-10GE-LR	SFP+ 10GBASE-LR 10 Gigabit Ethernet Optics, 1,310 nm for 10 km transmission on single mode fiber-optic (SMF)
QFX-SFP-10GE-ER	SFP+ 10GBASE-ER 10 Gigabit Ethernet Optics, 1,550 nm for 40 km transmission on single-mode fiber (SMF)
EX-SFP-10GE-ZR	SFP+ 10GBASE-ZR 10 Gigabit Ethernet Optics, 1,550 nm for 80 km transmission on single-mode fiber (SMF)
QFX-SFP-DAC-1M	SFP+ 10 Gigabit Ethernet Direct Attach Copper (twinax copper cable) 1 m
QFX-SFP-DAC-3M	SFP+ 10 Gigabit Ethernet Direct Attach Copper (twinax copper cable) 3 m
QFX-SFP-DAC-5M	SFP+ 10 Gigabit Ethernet Direct Attach Copper (twinax copper cable) 5 m
QFX-SFP-DAC-1MA	SFP+ 10 Gigabit Ethernet Direct Attach Copper (active twinax copper cable) 1 m
QFX-SFP-DAC-3MA	SFP+ 10 Gigabit Ethernet Direct Attach Copper (active twinax copper cable) 3 m
QFX-SFP-DAC-5MA	SFP+ 10 Gigabit Ethernet Direct Attach Copper (active twinax copper cable) 5 m
QFX-SFP-DAC-7MA	SFP+ 10 Gigabit Ethernet Direct Attach Copper (active twinax copper cable) 7 m
QFX-SFP-DAC-10MA	SFP+ 10 Gigabit Ethernet Direct Attach Copper (active twinax copper cable) 10 m
QFX-QSFP-DAC-1M	QSFP+ to QSFP+ Ethernet Direct Attach Copper (twinax copper cable) 1m passive
QFX-QSFP-DAC-3M	QSFP+ to QSFP+ Ethernet Direct Attach Copper (twinax copper cable) 3m passive
JNP-QSFP-DAC-5M	QSFP+ to QSFP+ Ethernet Direct Attach Copper (twinax copper cable) 5m passive
QFX-QSFP-DACBO-1M	QSFP+ to SFP+ 10 Gigabit Ethernet Direct Attach Breakout Copper (twinax copper cable) 1m
QFX-QSFP-DACBO-3M	QSFP+ to SFP+ 10 Gigabit Ethernet Direct Attach Breakout Copper (twinax copper cable) 3m
QFX-QSFP-40G-SR4	QSFP+ 40GBASE-SR4 40 Gigabit Optics, 850 nm for up to 150 m transmission on MMF

Model Number	Description
QFX-QSFP-40G-ESR4	QSFP+ 40GBASE-ESR4 40 Gigabit Optics, 300 m(400 m) with OM3(OM4) MMF
JNP-QSFP-40G-LR4	QSFP+ 40GBASE-LR4 40 Gigabit Optics, 1310nm for up to 10km Transmission on SMF
QFX-QSFP-DACBO-1M	QSFP+ to SFP+ 10 Gigabit Ethernet Direct Attach Break out Copper (twinax copper cable) 1 m
QFX-QSFP-DACBO-3M	QSFP+ to SFP+ 10 Gigabit Ethernet Direct Attach Break out Copper (twinax copper cable) 3 m
QFX-SFP-1GE-T	SFP 1000BASE-T Copper Transceiver Module for up to 100 m transmission on Cat5
QFX-SFP-1GE-SX	SFP 1000BASE-SX Gigabit Ethernet Optics, 850 nm for up to 550 m transmission on MMF
QFX-SFP-1GE-LX	SFP 1000BASE-LX Gigabit Ethernet Optics, 1,310 nm for 10 km transmission on SMF
JNP-QSFP-40G-LX4	QSFP+ 40GBASE-LX4 40 Gigabit Optics, 100m(150m) with OM3(OM4) duplex MMF

11 Appendices

This section contains the appendices that accompany the Security Target and provide clarity and/or explanation for the reader.

11.1 References

[800-56A] NIST Special Publication 800-56A, Revision 2, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1 Revision 3, July 2009, CCMB-2009-07-001.

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1 Revision 3, July 2009, CCMB-2009-07-002.

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, Version 3.1 Revision 3, July 2009, CCMB-2009-07-003.

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 3, July 2009, CCMB-2009-07-004.

[ECG14.1] Junos OS Common Criteria Evaluation Configuration Guide for QFX5100 and EX4600 Devices Release 14.1X53-D30

[FIPS140] Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, May 25, 2001. (Change notice (12-03-2002))

[FIPS197] Federal Information Processing Standard Publication (FIPS-PUB) 197, Advanced Encryption Standard (AES), November 2001.

[NDPP] Security Requirements for Network Devices, Version 1.1, 08 June 2012

[NDPPerr] Security Requirements for Network Devices Errata #3, 4 November 2014

[RFC4251] Internet Engineering Task Force, The Secure Shell (SSH) Protocol Architecture, January 2006

[RFC4252] Internet Engineering Task Force, The Secure Shell (SSH) Authentication Protocol, January 2006

[RFC4253] Internet Engineering Task Force, The Secure Shell (SSH) Transport Layer Protocol, January 2006

[RFC4254] Internet Engineering Task Force, The Secure Shell (SSH) Connection Protocol, January 2006

[RFC5656] Internet Engineering Task Force, Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer, December 2009

[RFC6668] Internet Engineering Task Force, SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol, July 2012

[SG_EX] Complete Software Guide for Junos OS for EX4600 Ethernet Switches, Release 14.1X53, Published 2015-02-22

[SG_QFX] Complete Software Guide for Junos OS for QFX Series Switches, Release 14.1X53d30

11.2 Glossary

Access – Interaction between an entity and an object that results in the flow or modification of data.

Access Control – Security service that controls the use of resources and the disclosure and modification of data.

Administrator – A user who has been specifically granted the authority to manage some portion or the entire TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance – A measure of confidence that the security features of an IT system are sufficient to enforce its security policy.

Asymmetric Cryptographic System – A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

Asymmetric Key – The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system

Attack – An intentional act attempting to violate the security policy of an IT system.

Authentication – Security measure that verifies a claimed identity.

Authentication data – Information used to verify a claimed identity.

Authorization – Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user – An authenticated user who may, in accordance with the TSP, perform an operation.

Compromise – Violation of a security policy.

Confidentiality – A security policy pertaining to disclosure of data.

Critical Security Parameters (CSP) – Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic boundary – An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) – A parameter used in conjunction with a cryptographic algorithm that determines:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a digital authentication code computed from data.

Cryptographic Module – The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Cryptographic Module Security Policy – A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

Entity – A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

External IT entity – Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

HCM JVAS – HTTP Content Management Juniper Value Added Software. This is an application used to inspect HTTP traffic; irrespective of the port on which the HTTP traffic arrives (i.e. it is not bound to port 80). Although use of this application is out of scope of the evaluation, the daemon supporting the service cannot be disabled. This service can only inspect http transit traffic, and cannot be used to undermine the configuration or operation of the TOE.

Identity – A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity – A security policy pertaining to the corruption of data and TSF mechanisms.

JUNOScope – A management framework that consists of tools for managing IP services for EX-Series Ethernet Switches. Use of JUNOScope is not supported in the evaluated configuration.

JUNOScript – An XML-based API for managing devices, developed by Juniper Networks. Use of JUNOScript is not supported in the evaluated configuration.

Junos OS VM – A virtual machine that executes in a CentOS virtualized environment.

Mandatory Access Control (MAC) – A means of restricting access to objects based on subject and object sensitivity labels.

Object – An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment – The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) – An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Security attributes – TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.

Security level – The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.

Sensitivity label – A security attribute that represents the security level of an object and that describes the sensitivity (e.g., Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decision.

Subject – An entity within the TSC that causes operation to be performed.

Symmetric key – A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

Threat – Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent – Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

User – Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability – A weakness that can be exploited to violate the TOE security policy.

11.3 Acronyms

TERM	DEFINITION
AES	Advanced Encryption Standard
API	Application Program Interface
CC	Common Criteria
CCMB	Common Criteria Management Board
CM	Configuration Management
CSP	Cryptographic security parameter
DES	Data Encryption Standard
DH	Diffie Hellman
DPC	Dense Port Concentrator
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
FIPS-PUB 140-2	Federal Information Processing Standard Publication
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
HCM	HTTP Content Management
HMAC	Keyed-Hash Authentication Code

TERM	DEFINITION
HTTP	Hypertext Transfer Protocol
ID	Identification
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IT	Information Technology
Junos	Juniper Operating System
JVAS	Juniper Value-Added Software
MAC	Mandatory Access Control
NDPP	Network Devices Protection Profile
NIAP	National Information Assurance Program
NIST	National Institute of Standards Technology
OAM	Operations, Administration and Maintenance
OSP	Organizational Security Policy
PAM	Pluggable Authentication Module
PFE	Packet Forwarding Engine
PIC	Physical Interface Card
PP	Protection Profile
RE	Routing Engine
RFC	Request for Comment
RNG	Random Number Generator
RNG	Random Number Generator
RSA	Rivest, Shamir, Adelman
SFR	Security Functional Requirement
SFP When used in description of Junos_	Small Form-factor Pluggable transceiver
SFP (when used In SFR)	Security Functional Policy
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Function
TSFI	TSF interfaces
TSP	TOE Security Policy
UDP	User Datagram Protocol
VM	Virtual Machine

Table 17 - Acronyms used in the Security Target