Security Target

Juniper Networks M-Series Multiservice Edge Routers, MX-Series 3D Universal Edge Routers, T-Series Core Routers and EX-Series Ethernet Switches Running Junos 12.1R3.6

ST Version1.7

May 12, 2014

Prepared By:

Juniper Networks, Inc.

1194 North Mathilda Avenue

Sunnyvale, CA 94089

www.juniper.net

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the M-Series Multiservice Edge Routers, MX-Series 3D Universal Edge Routers, T-Series Core Routers and EX-Series Ethernet Switches running Junos 12.1R3.6. 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 M-Series Multiservice Edge Routers, MX-Series 3D Universal Edge Routers, T-Series Core Routers and EX-Series Ethernet Switches running Junos 12.1R3.6
ST Revision	1.7
ST Draft Date	May 12, 2014
Author	Juniper Networks, Inc.

1.2 TOE Reference

TOE Reference	Juniper Networks M-Series Multiservice Edge Routers, MX-Series 3D Universal Edge Routers, T-Series Core Routers and EX-Series Ethernet Switches running Junos 12.1R3.6

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 PP 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 8.2for the Glossary.

1.4 TOE Overview

The Target of Evaluation (TOE) includes the following secure network devices running Junos 12.1R3.6:

- M-Series Multiservice Edge Routers:M7i M10i M120 M320
- MX-Series 3D Universal Edge Routers: MX5 MX10 MX40 MX80MX240 MX480 MX960
- T-Series Core Routers:T320 T640 T1600
- EX-Series Ethernet Switches:EX3300 EX4200 EX4500 EX6200 EX8200

[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.

1.5 TOE Boundaries

The TOE consists of the following components:

1. Network devices (as detailed in Table 2 below).
2. Junos 12.1R3.6: an operating system for security appliances.

The TOE is managed and configured via Command Line Interface.

Each appliance is a secure network device that protects itself largely by offering only a minimal logical interface to the network and attached nodes. All router platforms are powered by the same JUNOS software, which is a special purpose OS that provides no general purpose programming capability. JUNOS provides both management and control functions as well as all IP routing.

Each Juniper Networks M-series, T-series and MX-series routing platform is a complete routing system that supports a variety of high-speed interfaces (only Ethernet is within scope of the evaluation) for medium/large networks and network applications. Juniper Networks routers share common JUNOS software, features, and technology for compatibility across platforms.

The hardware has two components: the router itself and the PICs/DPC that have been installed in the router. The various PICs/DPC that have been installed in the router allow it to communicate with the different types of networks that may be required within the environment where the router will be used.

The router architecture of each platform cleanly separates routing and control functions from packet forwarding operations, thereby eliminating bottlenecks and permitting the router to maintain a high level of performance.

Each router consists of two major architectural components:

- The Routing Engine (RE), which provides Layer 3 routing services and network management and control;
- The Packet Forwarding Engine (PFE), 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.

The EX-series platforms provide high-performance, carrier-class networking solutions, supporting a variety of high-speed Ethernet interfaces for medium/large networks. The EX-series platforms share common JUNOS software with the routers, such that control plane features are implemented consistently with those of the routers.

The EX-series platforms are designed as hardware devices, featuring complete Layer 2 and Layer 3 switching capabilities. The EX-series platforms are powered by the same JUNOS modular architecture as the routers. The hardware abstraction layer allows control-plane features to be written once and implemented seamlessly on the underlying hardware. This modular approach also enhances fault-tolerance, as each JUNOS software protocol daemon runs in its own protected memory space and can be gracefully restarted independently without impacting the rest of the system.

1.5.1 Physical Boundary

M-Series	M7i M10i M120 M320
MX-Series	MX5 MX10 MX40 MX80MX240 MX480 MX960
T-Series	T320 T640 T1600
EX-Series	EX3300 EX4200 EX4500 EX6200 EX8200

Table 2 - List of Network Device Hardware

The TOE is comprised of appliance chassis (appliances listed in Table 2 above) and the Junos 12.1R3.6software and firmware running on the appliance (including the software implementing the Routing Engine and the software and ASICs implementing the Packet Forwarding Engine[2]). Hence the TOE is contained within the physical boundary of the specified appliance chassis.

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 7. 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/Information Flow Control	The TOE is designed to forward network packets (i.e., information flows) from source network entities to destination network entities based on available routing information. This information is either provided directly by TOE users or indirectly from other network entities (outside the TOE) configured by the TOE users. The TOE has the capability to regulate the information flow across its interfaces; traffic filters can be set in accordance with the presumed identity of the source, the identity of the destination, the transport layer protocol, the source service identifier, and the destination service identifier (TCP or UDP port number).
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

[2]The lower layers of the PFE (the DPC, PIC and Line Card network interface components) which simply deal with physical interfaces mechanics are out of scope.

TSF	DESCRIPTION
	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; • 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 Summary of Out-of-Scope Items

The only security functionality addressed by the evaluation is the functionality specified by the functional requirements in Section 5.1, and does not include additional product capabilities such as use of IPsec and information flow control based on traffic filters. The following items are out of the scope of the evaluation:

- External syslog server[3]
- Use of telnet, since it violates the Trusted Path requirement set (see Section 5.1)
- Use of FTP, since it violates the Trusted Path requirement set (see Section 5.1)
- Use of SNMP, since it violates the Trusted Path requirement set (see Section 5.1)
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set (see Section 5.1)
- Media use (other than during installation of the TOE)
- Use of root account, other than during initial installation and configuration.

[3]Although an external syslog server is expected to be present in the operational environment, the syslog server itself is not subject to evaluation.

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 to the following Protection Profile:

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

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 requirements in 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.

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 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.

5.1 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

5.1.1 Security Audit (FAU)

5.1.1.1 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 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 7].

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 16, Section 7].

5.1.1.2 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.

5.1.1.3 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.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 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,]

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

5.1.2.2 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.

5.1.2.3 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, 256-bits, and [192 bits] that meets the following:

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

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

FCS_COP.1.1(2) The TSF shall perform cryptographic signature services in accordance with [RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater], that meets the following:

- [FIPS PUB 186-2, "Digital Signature Standard "]

5.1.2.5 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] and message digest sizes [160, 256] bits that meet the following: FIPS Pub 180-3, "Secure Hash Standard."

5.1.2.6 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], key size [*160, 256 bits*], and message digest sizes [160, 256] bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

5.1.2.7 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 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 bit length of the keys and authorization factors that it will generate.

5.1.2.8 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.

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 [*32768*] 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 SSH_RSA 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].

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.

5.1.3 User Data Protection (FDP)

5.1.3.1 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 [deallocation of the resource from] all objects.

5.1.4 Identification and Authentication (FIA)

5.1.4.1 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**[4] Administrator, and support passwords of 15 characters or greater;

5.1.4.2 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;
- [[*ping, arp services*]].

[4]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 5.1.5.3). Therefore, the ST has adopted and applied the term “Authorized Administrator” for consistency reasons.

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.

5.1.4.3 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.

5.1.4.4 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.

5.1.5 Security Management (FMT)

5.1.5.1 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**[4] Administrators.

5.1.5.2 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;
- [No other capabilities].

5.1.5.3 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.

5.1.6 Protection of the TSF (FPT)

5.1.6.1 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.

5.1.6.2 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.

5.1.6.3 Reliable time stamps (FPT_STM.1)

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

5.1.6.4 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide **authorized**[4] administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide **authorized**[4] 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.

5.1.6.5 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.

5.1.7 TOE Access (FTA)

5.1.7.1 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**[4]Administrator-specified time period of inactivity.

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

FTA_SSL.3.1 The TSF shall terminate a remote interactive session after a [after **anAuthorized**[4]Administrator-specified time period of inactivity].

5.1.7.3 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.

5.1.7.4 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing an administrative user session the TSF shall display **an Authorized**[4]Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.8 Trusted Path/Channels (FTP)

5.1.8.1 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*].

5.1.8.2 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.

5.2 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.1	Security objectives
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	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

5.3 Security Requirements Rationale

5.3.1 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.1which authenticates administrator identity prior to interaction with TSF.
FAU_STG_EXT.1	Assumed to be: FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel	FAU_GEN.1, FTP_ITC.1
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	Assumed to be: [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]	FDP_ITC.1, 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

SFR	Dependency	Satisfaction of dependency
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	Assumed to be: None	n/a
FCS_SSH_EXT.1	Assumed to be: FCS_COP.1	FCS_COP.1(1-4)
FDP_RIP.2	None	n/a
FIA_PMG_EXT.1	Assumed to be: FIA_UAU_EXT.2 Password-based authentication mechanism	FIA_UAU_EXT.2
FIA_UIA_EXT.1	Assumed to be: None	n/a
FIA_UAU_EXT.2	Assumed to be: FIA_PMG_EXT.1 Password management	FIA_PMG_EXT.1
FIA_UAU.7	FIA_UAU.1 Timing of authentication	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FMT_MTD.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FMT_SMF.1	None	n/a
FMT_SMR.2	FIA_UID.1 Timing of identification	FIA_UIA_EXT.1 which authenticates administrator identity prior to interaction with TSF.
FPT_SKP_EXT.1	Assumed to be: None	n/a

SFR	Dependency	Satisfaction of dependency
FPT_APW_EXT.1.1	Assumed to be: None	n/a
FPT_STM.1	None	n/a
FPT_TUD_EXT.1	Assumed to be: None	n/a
FPT_TST_EXT.1	Assumed to be: None	n/a
FTA_EXT_SSL.1	Assumed to be: FIA_UIA_EXT.1 User identification and authentication	FIA_UIA_EXT.1
FTA_SSL.3	None	n/a
FTA_SSL.4	None	n/a
FTA_TAB.1	None	n/a
FTP_ITC.1	None	n/a
FTP_TRP.1	None	n/a

Table 11– Satisfaction of dependencies

5.3.2 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].

6 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.

6.1 Security Audit

JUNOS 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;
b) Configuration is committed;
c) Configuration is changed;
d) All use of the identification and authentication mechanisms;
e) Service requests;
f) Failure to establish an SSH session establishment/termination of an SSH session;
g) Changes to the time;
h) Initiation of update;
i) Indication that TSF self-test was completed;
j) Termination of a remote session by the session locking mechanism;
k) Termination of an interactive session;
l) Initiation/termination/failure of the trusted channel functions.

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[5]). Local audit log are stored in /var/log/. Only an authorized administrator can read log files, or delete log and archive files. The syslog files 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 *Junos OS System Basics Configuration Guide* [SB] Chapter 9, Subsection 'Specifying Log File Size, Number, and Archiving Properties'). 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 further details see *Junos OS System Basics Configuration Guide* [SB] Chapter 13, Subsection "archive (All System Log Files)".

The maximum value that can be specified for the size of a log file is 1GB. However, the default maximum size depends on the platform type:

- 1 megabyte (MB) for M Series, MX Series, and T Series routers
- 128KB for the EX Series switches

These defaults maximum sizes can be modified by the user, as detailed in *Junos OS System Basics Configuration Guide* [SB] Chapter 9, Subsection 'Specifying Log File Size, Number, and Archiving Properties'.

[5] In accordance with RFC 4741.

For more information about configuring event logging, see the *Junos OS System Basics Configuration Guide* [SB] and the *Junos OS Common Criteria Evaluated Configuration Guide for* EX Series, M Series, MX Series, and T series *Devices 12.1R3.6* [ECG].

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

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

6.2 Cryptographic Support

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

Implementation	Algorithm	Cert Number	Cert Number
		M,T,Mx	**EX**
RNG	N/A	1118	1120
MD	SHA	1921	1925
	HMAC	1361	1366
Kernel	AES	2232	2236
	SHA	1920	1924
	HMAC	1360	1365
OpenSSL	AES	2233	2237
	RSA	1144	1146
	ECDSA	346	348
	SHA	1922	1926
	HMAC	1362	1367

Table 12 - CAVS Certificate Results

The TOE meets the cryptographic requirements either by allowing the administrator to enable the FIPS operating mode or by running a FIPS image (per platform guidance). The evaluated configuration of the TOE requires the use of this FIPS operating mode. The Cryptographic security function is described in the context of how it satisfies the cryptographic security requirements.

The FIPS-approved cryptomodule implements RSA Digital Signature Standard using a base point of 2048-bits or greater (as specified by the cryptographic administrator) 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 32768 bytes 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 "SSH-RSA" as its public key algorithm.

The data integrity algorithms used in SSH transport connection is”hmac-sha1", as required by [RFC4253].

Key exchange is done using "diffie-hellman-group14-sha1" [RFC4253].

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

The TOE performs random number generation in accordance with NIST Special Publication 800-90 using HMAC_DRBG, SHA-256

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	Overwritten three times, first with the byte pattern 0xff, then 0x00, and then 0xff again, before they are deleted
SSH Session Key	Session keys used with SSH, AES 128, 256, HMAC-SHA-1 key (160), DH Private Key 1024	Plaintext	Memory	Scrubbed in memory using OpenSSL scrubbing method, overwriting the buffer with random data
User Password	Plaintext value as entered by user	Hashed	Memory	Overwritten with zero's
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's

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

6.3 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 OS is aware of when the end of the packet is reached and pads a short packet with zeros accordingly. Therefore, 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

6.4 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”. Locally stored authentication data for fixed password authentication is a case-sensitive, alphanumeric value. The password has a minimum length of 15[6] characters with at least two changes of character set (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

- 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 6.8), 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

[6]By default the minimum password length is 10, but this should be set to minimum length of 15 in the evaluated configuration using the command:set system login password minimum-length 15

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. 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()[7] 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 router. The TOE successfully authenticates if the authentication data provided matches that stored in conjunction with the provided identity.

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

6.5 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, alphanumeric value. 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 router. 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/'). Only the authentication subsystem has the capability to decrypt the password. 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

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

- Restrict the service available to unidentified or unauthenticated IT entities
- Restrict TOE (release) updates[8]

Detailed topics on the secure management of Juniper's routers & switches are discussed in the *Junos OS System Basics Configuration Guide* [SB] and the *Junos OS Common Criteria Evaluated Configuration Guide for EX Series, M Series, MX Series, and T series Devices OS 12.1R3.6* [ECG].

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

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

6.6 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. The clock function is reliant on the system clock provided by the underlying hardware. In addition, 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.

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 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).

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 addjinstall), 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.

The TOE will run the following set of self tests during power on to check the correct operation 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.

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

- Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.

The power on self-tests may produce some or all of the output shown in Figure 1 - FIPs Self-Test Example, below.

```
request system fips self-test
user@switch> request system fips self-test Testing file integrity:
File integrity Known Answer Test:                  Passed
Testing crypto integrity:
   Crypto integrity Known Answer Test:             Passed
Testing kernel KATS:
   DES3-CBC Known Answer Test:                     Passed
   HMAC-SHA1 Known Answer Test:                    Passed
   HMAC-SHA2-256 Known Answer Test:                Passed
   SHA-2 Known Answer Test:                        Passed
   AES128-CMAC Known Answer Test:                  Passed
   AES-CBC Known Answer Test:                      Passed
Testing libmd KATS:
   HMAC-SHA1 Known Answer Test:                    Passed
   HMAC-SHA2-256 Known Answer Test:                Passed
Testing OpenSSL KATS:
   FIPS RNG Known Answer Test:                     Passed
   FIPS DSA Known Answer Test:                     Passed
   FIPS ECDSA Known Answer Test:                   Passed
   FIPS ECDH Known Answer Test:                    Passed
   FIPS RSA Known Answer Test:                     Passed
   DES3-CBC Known Answer Test:                     Passed
   HMAC-SHA1 Known Answer Test:                    Passed
   SHA-2 Known Answer Test:                        Passed
   AES-CBC Known Answer Test:                      Passed
   ECDSA-SIGN Known Answer Test:                   Passed
   KDF-IKE-V1 Known Answer Test:                   Passed
Testing SSH IPsec KATS:
   DES3-CBC Known Answer Test:                     Passed
   HMAC-SHA1 Known Answer Test:                    Passed
   HMAC-SHA2-256 Known Answer Test:                Passed
   SHA-2 Known Answer Test:                        Passed
   AES-CBC Known Answer Test:                      Passed
   SSH-RSA-ENC Known Answer Test:                  Passed
   SSH-RSA-SIGN Known Answer Test:                 Passed
   KDF-IKE-V1 Known Answer Test:                   Passed
Expect an exec Authentication error...
exec: /opt/sbin/kats/cannot-exec.real: Authentication error
```

Figure 1 - FIPs Self-Test Example

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. This automatic recovery and self-test behavior, is discussed in Chapter 7 of the *Junos OS Evaluated Configuration Guide for EX Series, M Series, MX Series, and T series Devices OS 12.1R3.6* [ECG].

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.

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

6.7 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 router as well as any other information that the Authorized Administrator wishes to communicate.

User sessions can be locked or 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 locking and 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

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

- FTP_ITC.1
- FPT_TRP.1

6.9 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 has one RSA, one DSA, and one ECDSA Host Key for SSH v2, which are generated on initial setup of the TOE. Any of them can be deconfigured via the CLI and the relevant key will be deleted and thus unavailable during connection establishment. These keys are 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. The TOE can be configured with ACLs to control the clients that are able to connect to it via SSH. **Ordering of Key Exchange Methods:** The TOE orders key exchange algorithms as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1, diffie-hellman-group1-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-ctr,aes192-ctr,aes256-ctr,aes128-cbc,aes192-cbc,aes256-cbc, and 3des-cbc[9]. 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 MAC algorithms in 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 32768 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.

Table 14 – RFC Conformance Statements

[9]Others are supported by default, but these are the encryption algorithms the Secure Configuration Guide specifies are to be configured in the evaluated configuration.

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

6.10 800-56 Conformance Statements

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.

6.10.1 Finite Field-Based Key Establishment Schemes

The requirements for Finite Field-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. The FFC parameter is set and so ECC is not used
	5.5.1.2 ECC Domain Parameter Generation	n/a – Elliptic Curve Cryptography is not used.
5.6 Private and Public Keys	General	No statements
5.6.1 Private/Public Key Pair Generation	5.6.1.1 FFC Key Pair Generation	Comply with all "shall" statements. Only static public keys used.
	5.6.1.2 ECC Key Pair Generation	n/a – Elliptic Curve Cryptography is not used.
5.6.2 Assurances of the Arithmetic Validity of a Public Key	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 Owner Assurances of Static Public Key Validity	Owner Full Validation - The owner performs a successful full public key validation, via pair-wise consistency check

800-56A section	800-56A sub section	Compliance
	5.6.2.2 Recipient Assurances of Static Public Key Validity	TTP Generation – 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.
	5.6.2.3 Recipient Assurances of Ephemeral Public Key Validity	Recipient Full Validation - The recipient performs a successful full public key Validation.
	5.6.2.4 FFC Full Public Key Validation Routine	Comply with “shall” statement.
	5.6.2.5 ECC Full Public Key Validation Routine	n/a – Elliptic Curve not used.
	5.6.2.6 ECC Partial Public Key Validation Routine	n/a – Elliptic Curve not used.
5.6.3 Assurances of the Possession of a Static Private Key	General	Comply with “shall” statement.
	5.6.3.1 Owner Assurances of Possession of a Static Private Key	Owner Receives Assurance via Key Generation - The act of generating a key pair.
5.6.3.2 Recipient Assurance of Owner’s Possession of a Static Private Key	General	Comply with all “shall” statements.
	5.6.3.2.1 Recipient Obtains Assurance through a Trusted Third Party	The TOE will be made aware of the method(s) used by the third party.
	5.6.3.2.2 Recipient Obtains Assurance Directly from the Claimed Owner	The underlying key agreement used by the TOE is “dhOneFlow or (Cofactor) One-Pass Diffie-Hellman”. Comply with all “shall” statements.
5.6.4 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.

800-56A section	800-56A sub section	Compliance
	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

6.10.2 Elliptic Curve Cryptography-Based Schemes (ECC Scheme)

Elliptic curve is not implemented in the TOE.

7 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.1	None	
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 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.
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

8 Appendices

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

8.1 References

[AES] The AES Cipher Algorithm and Its Use with IPsec <draft-ietf-ipsec-ciph-aes-cbc.03.txt>, Internet draft, November 2001.

[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.

[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

[RFC2451] Internet Engineering Task Force, ESP CBC-Mode Cipher Algorithms, RFC 2451, November 1998.

[RFC2409] Internet Engineering Task Force, Internet Key Exchange (IKE), RFC 2409, November 1998.

[RFC2406] Internet Engineering Task Force, IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

[RFC2404] Internet Engineering Task Force, Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November 1998.

[SB] Junos OS System Basics Configuration Guide, Release 12.1, Published 2012-05-08

[ECG] Junos OS Common Criteria Evaluated Configuration Guide for EX Series, M Series, MX Series, and T series Devices Junos OS 12.1R3.6.

8.2 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.

Accountability – Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Active – (scanning capability) – to gain understanding of the IT environment through means that illuminate the environment being scanned.

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 it's' 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.

Availability – Timely, reliable access to IT resources.

Component – A single scanning capability, sensing capability or analyzing capability, operating within the TOE configuration

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.

Defense-in-Depth (DID) – A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) – A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. Those controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Embedded Cryptographic Module – On that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

Enclave – A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

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.

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.

Integrity level – The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Intrusion – Any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource.

Intrusion Detection – Pertaining to techniques which attempt to detect intrusion into an IT System by observation of actions, security logs, or audit data. Detection of break-ins or attempts either manually or via software expert systems that operate on logs or other information available on the network.

Intrusion Detection System (IDS) – A combination of one or more sensing capabilities, and one or more analyzing capabilities and an optional but recommended scanning capability that monitor an IT System for activity that may inappropriately affect the IT System's assets and react appropriately.

Intrusion Detection System Analyzing Capability – The components of an IDS that accepts data from sensing capabilities and scanning capabilities and other IT System resources, and then applies analytical processes and information to derive conclusions about intrusions (past, present, or future).

Intrusion Detection System Data (IDS data) – Data collected and produced by the IDS functions. This could include digital signatures, policies, permissions, and IDS audit data.

Intrusion Detection System Sensing Capability – The component of an IDS that collects real-time events that may be indicative of vulnerabilities in or misuse of IT resources.

Jade – A daemon used for the authentication of JUNOScript connections.

JUNOScope – A management framework that consists of tools for managing IP services for the M-Series Multiservice Edge Routers, MX-Series 3D Universal Edge Routers, T-Series Core Routers and 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.

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

Mandatory Integrity Control (MIC) – A means of restricting access to objects based on subject and object integrity labels.

Multilevel – The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object – An object that exhibits all of the following characteristics:

The object may be used to transfer information between subjects of differing user identities within the TSF.

Subjects in the TOE must be able to require a specific instance of the object.

The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to require the same instance of the object.

Non-Repudiation – A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

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.

Operational key – Key intended for protection of operational information or for the production or secure electrical transmissions of key streams

Passive – (sensing capability) – To gain understanding of the IT environment through means that do not effect or impact the environment being sensed.

Peer TOEs – Mutually authenticated TOEs that interact to enforce a common security policy.

Public Object – An object for which the TSF unconditionally permits all entities “read” access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Release Train -- The technique of planning software releases on regular or cyclic time period, for example, the last day of every quarter, or every 9 weeks, etc. The "train" metaphor of a release train is

likely based on the concept of railroad train schedules (planned arrival and departure times) and that trains carry multiple types of rolling stock (different types of features are included in a release).

Robustness – A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness:

- Basic: Security services and mechanisms that equate to good commercial practices.
- Medium: Security services and mechanisms that provide for layering of additional safeguards above good commercial practices.
- High: Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

Secure State – Condition in which all TOE security policies are enforced.

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.

Split key – A variable that consists of two or more components that must be combined to form the operation key variable. The combining process excludes concatenation or interleaving of component variables.

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.

8.3 Acronyms

TERM	DEFINITION
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Program Interface
ATM	Asynchronous Transfer Method
BGP	Border Gateway Protocol
CC	Common Criteria version 3.1
CCEVS	Common Criteria Evaluation Validation Scheme
CCIMB	Common Criteria Interpretations Management Board
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CLNP	Connectionless Network Protocol
CLNS	Connectionless Network Service
CM	Configuration Management
CSP	Cryptographic security parameter
DES	Data Encryption Standard
DH	Diffie Hellman
DMZ	Demilitarized Zone
DoD	Department of Defense
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
GIG	Global Information Grid
GUI	Graphical User Interface
HMAC	Keyed-Hash Authentication Code
HTTP	Hypertext Transfer Protocol
I&A	Identification and Authentication
IATF	Information Assurance Technical Framework
ICMP	Internet Control Message Protocol
ID	Identification
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange

TERM	DEFINITION
IP	Internet Protocol
IPsec	Internet Protocol Security
IPsec ESP	Internet Protocol Security Encapsulating Security Payload
IPv6	Internet Protocol Version 6
IPX	Internetwork Packet Exchange
ISAKMP	Internet Security Association and Key Management Protocol
IS-IS	Intermediate System-to-Intermediate System
ISO	International Organization for Standardization
IT	Information Technology
Junos	Juniper Operating System
LDP	Label Distribution Protocol
MAC	Mandatory Access Control
MRE	Medium Robustness Environment
NAT	Network Address Translation
NBIAT&S	Network Boundary Information Assurance Technologies and Solutions Support
NDPP	Network Devices Protection Profile
NIAP	National Information Assurance Program
NIST	National Institute of Standards Technology
NSA	National Security Agency
OSI	Open Systems Interconnect
OSP	Organizational Security Policy
OSPF	Open Shortest Path First
PAM	Pluggable Authentication Module
PFE	Packet Forwarding Engine
PIC/PIM	Physical Interface Card/Module
PKI	Public Key Infrastructure
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RE	Routing Engine
RFC	Request for Comment
RIP	Routing Information Protocol
RNG	Random Number Generator
RNG	Random Number Generator
RSA	Rivest, Shamir, Adelman
SA	Security Association

TERM	DEFINITION
SCEP	Simple Certificate Enrollment Protocol
SFP	Security Functional Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOF	Strength of Function
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TCP/IP	Transmissions Control Protocol/ Internet Protocol
TDEA	Triple Data Encryption Algorithm
TFTP	Trivial File Transfer Protocol
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Function
TSFI	TSF interfaces
TSP	TOE Security Policy
TTAP/CCEVS	Trust Technology Assessment Program/ Common Criteria Evaluation Standard Scheme
UDP	User Datagram Protocol
URL	Uniform Research Locator
VPN	Virtual Private Network

Table 17 - Acronyms used in the Security Target