Security Target

Juniper Networks, Inc. Junos 12.1X46-D20 for SRX and LN Series Platforms (NDPP, TFFWEP, VPNEP)

Document Version 1.9

June 10, 2015

Prepared For:

Juniper Networks, Inc.

1194 North Mathilda Avenue

Sunnyvale, CA 94089

www.juniper.net

Prepared By:

Apex Assurance Group, LLC

530 Lytton Avenue, Ste. 200

Palo Alto, CA 94301

www.apexassurance.com

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Juniper Networks, Inc. Junos 12.1 X46 D20.6 for SRX and LN Series Platforms. 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.

Revision History

Revision	Date	Description
1.0	October 9, 2013	Initial version
1.1	March 14, 2014	Updated to address EOR
1.2	April 1, 2014	Updated to address EOR
1.3	July 8. 2014	Updated to address EOR 2.0
1.4	July 31, 2014	Updated TOE Reference
1.5	September 18, 2014	Updated to address EOR 3.0
1.6	November 16, 2014	Updated to address evaluator comments
1.7	January 21, 2015	Address certifier comments
1.8	May 7, 2015	PSK length update
1.9	June 10, 2015	Added CAVS Certificates numbers (Table 7-1 CAVS Certification Results)

Table of Contents

List of Tables

List of Figures

REMAINDER OF THIS PAGE INTENTIONALLY LEFT BLANK

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 product.

1.1 ST Reference

ST Title	Security Target: Juniper Networks, Inc. Junos 12.1X46-D20.6 for SRX and LN Series Platforms
ST Revision	1.9
ST Publication Date	June 10, 2015
Author	Apex Assurance Group, LLC

1.2 TOE Reference

TOE Reference	Juniper Networks, Inc. Junos 12.1X46-D20.6 for SRX and LN Series Platforms

1.3 Document Organization

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	Extended Components Definition	Describes extended components of the evaluation (if any)
6	Security Requirements	Contains the functional and assurance requirements for this TOE

SECTION	TITLE	DESCRIPTION
7	TOE Summary Specification	Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements.

Table 1-1 – ST Organization and Section Descriptions

1.4 Document Conventions

The notation, formatting, and conventions used in this Security Target are consistent with those used in Version 3.1 of the Common Criteria. Selected presentation choices are discussed here to aid the Security Target reader. The Common Criteria allows several operations to be performed on functional requirements: The allowable operations defined in Part 2 of the Common Criteria are *refinement, selection, assignment* and *iteration*.

- *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;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. When not embedded in a Security Functional Requirement, italicized text is used for both official document titles and text meant to be emphasized more than plain text.

1.5 Document Terminology

The following table describes the acronyms used in this document:

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

TERM	DEFINITION
DMZ	Demilitarized Zone
DoD	Department of Defense
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
FWEP	Firewall Extended Package
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
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
NTP	Network Time Protocol
OSI	Open Systems Interconnect
OSP	Organizational Security Policy
OSPF	Open Shortest Path First
PAM	Pluggable Authentication Module

TERM	DEFINITION
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
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
TBD	To Be Determined
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
VPNEP	Virtual Private Network Extended Package

Table 1-2 – Acronyms Used in Security Target

1.6 TOE Overview

The TOE is Juniper Networks, Inc. Junos 12.1 X46 D20.6 for SRX and LN Series Platforms which primarily supports the definition of and enforces information flow policies among network nodes. The routers

provide for stateful inspection of every packet that traverses the network and provide central management to manage the network security policy. All information flow from one network node to another passes through an instance of the TOE. Information flow is controlled on the basis of network node addresses, protocol, type of access requested, and services requested. In support of the information flow security functions, the TOE ensures that security-relevant activity is audited, that their own functions are protected from potential attacks, and provides the security tools to manage all of the security functions.

The Junos 12.1 X46 D20.6 for SRX and LN Series Platforms may also be referred to as the TOE in this document.

1.7 TOE Description

1.7.1 Overview

Each Juniper Networks routing platform is a complete routing system that supports a variety of high-speed interfaces (up to 10 Gbps) for medium/large networks and network applications. Juniper Networks routers share common JUNOS software, features, and technology for compatibility across platforms.

The routers are physically self-contained, housing the software, firmware and hardware necessary to perform all router functions. The hardware has two components: the router itself and various PIC/PIMs, which allow the routers to communicate with the different types of networks that may be required within the environment where the routers are used.

Each instance of the TOE consists of the following major architectural components:

- The Routing Engine (RE) runs the JUNOS software and provides Layer 3 routing services and network management for all operations necessary for the configuration and operation of the TOE and controls the flow of information through the TOE, including Network Address Translation (NAT) and all operations necessary for the encryption/decryption of packets for secure communication via the IPSec protocol.

- The Packet Forwarding Engine (PFE) 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 routers support numerous routing standards for flexibility and scalability as well as IETF IPSec protocols. These functions can all be managed through the JUNOS software, either from a connected terminal console or via a network connection. Network management can be secured using IPsec, SNMP

v3, and SSH protocols. All management, whether from a user connecting to a terminal or from the network, requires successful authentication.

Juniper Networks security devices accomplish routing through a process called a virtual router (VR). A security device divides its routing component into two or more VRs with each VR maintaining its own list of known networks in the form of a routing table, routing logic, and associated security zones. Note that virtual routers are not an evaluated feature.

The TOE is managed and configured via Command Line Interface using SSH connections and does not depend on FTP or or SSL to operate correctly.

1.7.2 Physical Boundary

The TOE is a combined hardware/software TOE and is defined as the Junos 12.1 X46 D20.6 for SRX and LN Series Platforms. The TOE boundary is shown below.

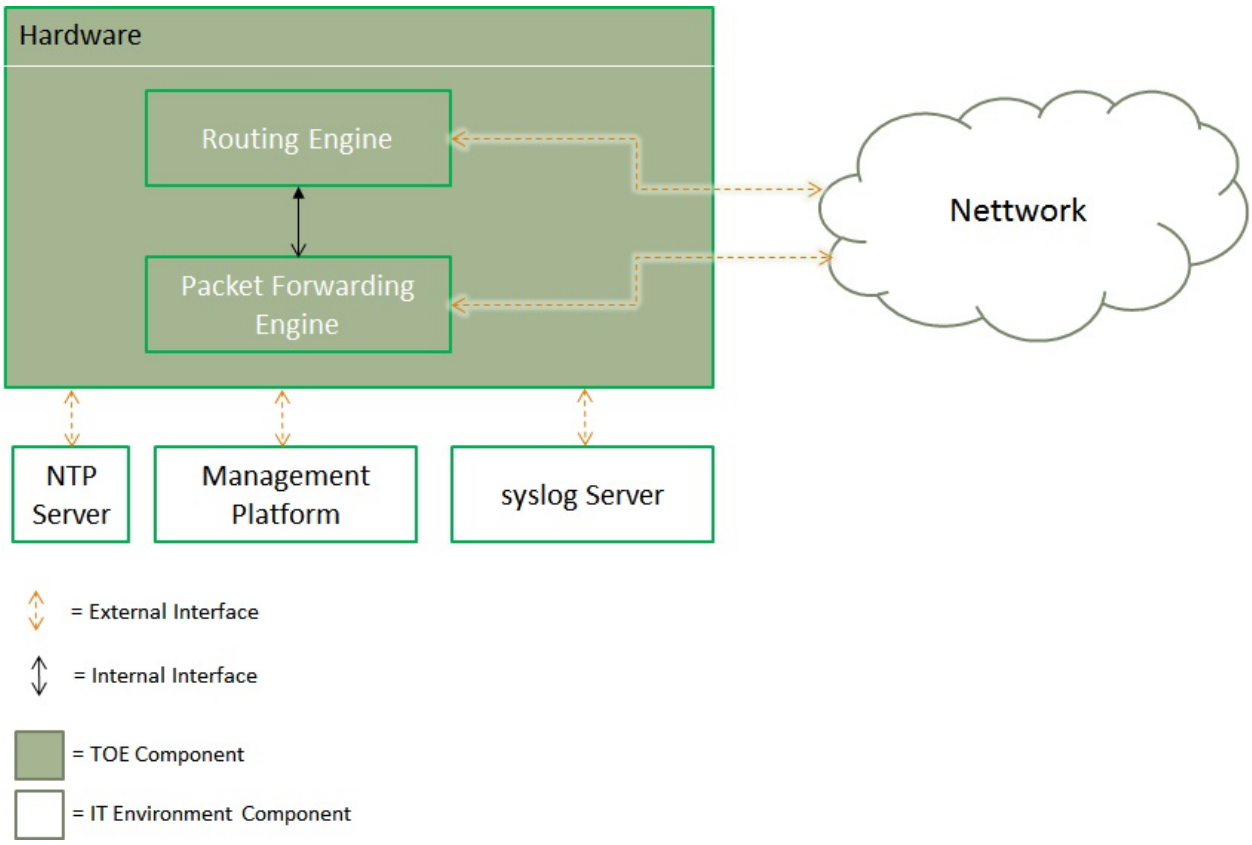

Figure 1 - TOE Boundary

The marketing name for the board that implements the PFE on the SRX 5000 series is the Services Processing Card (SPC). There are two models of the SPC available for the SRX 5000 series, but only one will be evaluated. No other SRX or LN models have a similar option.

The physical boundary is defined as the entire router chassis. In order to comply with the evaluated configuration, the following hardware and software components should be used:

TOE COMPONENT	VERSION/MODEL NUMBER
Software Version	Junos FIPS Version 12.1 X46 D20.6
Hardware Platforms	SRX100, SRX110, SRX210, SRX220, SRX240, SRX550 and SRX650; LN1000, LN2600 (same CPU and crypto processor as SRX650); SRX5400, SRX5600 and SRX5800 with SPC-4-15-320

Table 1-3 - Evaluated Configuration of the TOE

The TOE interfaces are comprised of the following:

1. Network interfaces which pass traffic
2. Management interface through which handle administrative actions.

1.7.3 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 table:

TSF	DESCRIPTION
Security Audit	JUNOS auditable events are stored in the syslog files, and can be sent to an external log server (via IPSec). 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 6. 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	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	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 Secure Shell (SSH) used to exchange information. Telnet, File Transfer Protocol (FTP), and Secure Socket Layer (SSL) are out of scope.

TSF	DESCRIPTION
Security Management	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	The TOE provides protection mechanisms TSF 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 by the TOE.
TOE Access	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	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.
Stateful Traffic/Packet Filtering	The TOE provides stateful network traffic filtering based on examination of network packets and the application of information flow rules.

Table 1-4 – Logical Boundary Descriptions

1.7.4 Summary of Out-of-Scope Items

The following items are out of the scope of the evaluation:

- External syslog server
- Use of telnet, since it violates the Trusted Path requirement set (see Security Requirements)
- Use of FTP, since it violates the Trusted Path requirement set (see Security Requirements)
- Use of SNMP, since it violates the Trusted Path requirement set (see Security Requirements)
- Management via J-Web, since it violates the Trusted Path requirement set (see Security Requirements)
- Media use (other than during installation of the TOE)
- Virtual Routers
- SSL

1.7.5 TOE Security Functional Policies

Since the NDPP, FWEP, and VPNEP do not require it, the TOE does not support any Security Functional Policy.

1.7.6 TOE Product Documentation

The TOE includes the following product documentation:

- *Junos OS Installation and Upgrade Guide Release 12.1 X46*
- *Junos OS CLI User Guide Release 12.1 X46*
- *Junos OS System Basics: Getting Started Configuration Guide Release 12.1 X46*
- *Common Criteria Evaluated Configuration Guide for LN Series Rugged Secure Routers and SRX Series Security Devices Published: 2014-10-12*
- *Annex for AGD (Assurance Guidance Document) Juniper Networks SRX Series Services Gateways Running Junos 12.1X46-D20 Version 1.1 October 8, 2014*

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 extended.

2.2 Protection Profile Conformance Claim

The TOE claims exact conformance to the following U.S. Government approved Protection Profiles (PP):

- Security Requirements for Network Devices, Version 1.1, 08 June 2012 (NDPP)
- Security Requirements for Network Devices Errata #2, 13 January 2013
- Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, Version 1.0, 19 December 2011 (FWEP)
- Network Device Protection Profile (NDPP) Extended Package VPN Gateway, Version 1.1, 12 April 2013 (VPNEP)

2.2.1 TOE Type Consistency

Both the PP and the TOE describe network device systems and firewalls.

2.2.2 Security Problem Definition Consistency

This ST claims exact conformance to the referenced PPs. The threats, assumptions, and organizational security policies in the ST are identical to the threats, assumptions, and organizational security policies in the PPs.

2.2.3 Security Objectives Consistency

This ST claims exact conformance to the objectives in the referenced PPs. No additions or deletions to the objectives have been made. All objectives are consistent with the PPs.

2.2.4 Security Functional Requirements Consistency

This ST claims exact conformance to the security functional requirements in the referenced PPs.

2.2.5 Security Assurance Requirements Consistency

This ST claims exact conformance to the security assurance requirements in the referenced PPs.

2.3 Package Claim

The TOE claims conformance to Security Requirements for Network Devices, Version 1.1, 08 June 2012 and the Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, Version 1.0, 19 December 2011 (FWEP), Network Device Protection Profile (NDPP) Extended Package VPN Gateway, Version 1.1, 12 April 2013 (VPNEP) and no other assurance or functional packages.

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], [FWEP] and [VPNEP}.

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], [FWEP], and [VPNEP] 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 administrator may unintentionally 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 3-1 – Threats from the NDPP addressed by the TOE

The following threats are addressed by the TOE, as detailed in section 5.1.2 of [FWEP].

THREAT	DESCRIPTION
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T. NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network.

THREAT	DESCRIPTION
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
T.NETWORK_DOS	Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network.

Table 3-2 - Threats from the FWEP addressed by the TOE

The following threats are addressed by the TOE, as detailed in section 2 of [VPNEP].

THREAT	DESCRIPTION
T.DATA_INTEGRITY	Known malicious external devices able to communicate with devices on the protected network or devices on the protected network establish communications with those external devices then the data contained within the communications may be susceptible to a loss of integrity.
T. REPLAY_ATTACK	Unauthorized individuals gains access to the system and may have the opportunity to conduct a “replay” attack.

Table 3-3 - Threats from the VPNEP not already included in FWEP

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	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 3-4 – 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.

ASSUMPTION	DESCRIPTION
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Table 3-5 – Assumptions from the NDPP

This section contains assumptions regarding the security environment and the intended usage of the TOE, as specified in section 5.1.1 of [FWEP].

ASSUMPTION	DESCRIPTION
A.CONNECTIONS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

Table 3-6 - Assumptions from the FWEP

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.

OBJECTIVES	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 4-1 – TOE Security Objectives from NDPP

The IT Security Objectives for the TOE are detailed below, as specified in section 5.2.1 of [FWEP].

OBJECTIVES	DESCRIPTION
O.ADDRESS_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination addresses.
O.PORT_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination transport layer ports.
O.STATEFUL_INSPECTION	The TOE will determine if a network packet belongs to an allowed established connection before applying the ruleset.
O.RELATED_CONNECTION_FILTERING	For specific protocols, the TOE will dynamically permit a network packet flow in response to a connection permitted by the ruleset.

Table 4-2 TOE Security Objectives from FWEP

The IT Security Objectives for the TOE are detailed below, as specified in section 3 of [VPNEP].

OBJECTIVES	DESCRIPTION

OBJECTIVES	DESCRIPTION
O.CRYPTOGRAPHIC_FUNCTIONS	The TOE will implement cryptographic capabilities to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.AUTHENTICATION	The TOE shall provide authentication ability (IPSec) to allow a VPN peer to establish VPN connectivity with another VPN peer. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.FAIL_SECURE	The TOE will shut down upon discovery of a problem reported via the self-test mechanism.

Table 4-3 TOE Security Objectives from VPNEP not already covered by FWEP or NDPP

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 administrator guidance in a trusted manner.

Table 4-4 – Operational Environment Security Objectives from NDPP.

The security objectives for the operational environment are detailed below, as specified in section 5.2.2 of [FWEP].

OBJECTIVE	DESCRIPTION
OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

Table 4-5 - Operational Environment Security Objectives from FWEP

4.3 Security Objectives Rationale

As these objectives for the TOE and operational environment are the same as those specified in [NDPP], [FWEP], and [VPNEP] The rationales provided in the prose of [NDPP] Section 3,in the tables in [NDPP] Annex A, section 5 of [FWEP] and section 3 of [VPNEP] 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], [FWEP], and [VPNEP].

5 Extended Components Definition

The following extended components are defined by the NDPP. The definition of these components is given in NDPP.

- FAU_STG_EXT.1
- FCS_CKM_EXT.4
- FCS_RBG_EXT.1
- FCS_SSH_EXT.1
- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.5
- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_TUD_EXT.1
- FPT_TST_EXT.1
- FTA_SSL_EXT.1

The following extended components are defined by the FWEP. The definition of these components is given in FWEP.

- FFW_RUL_EXT.1

The following extended components are defined by the VPNEP. The definition of these components is given in VPNEP.

- FCS_IPSEC_EXT.1
- FIA_PSK_EXT.1
- FIA_X509_EXT.1
- FPF_RUL_EXT.1

5.1 Rationale for Extended Components

This ST includes these extended components to conform to the NDPP, FWEP, and VPNEP requirements.

6 Security Requirements

The security requirements that are levied on the TOE and the Operational environment are specified in this section of the ST.

6.1 Security Functional Requirements

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

The following table identifies all the SFR's implemented by the TOE.

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	External Audit Trail Storage
Cryptographic Support	FCS_CKM.1(1), (2), (3)	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.1(2)	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), (3)	Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(4)	Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(5)	Cryptographic Operation (for keyed-hash message authentication)
	FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
	FCS_SSH_EXT.1	Explicit SSH
	FCS_IPSEC_EXT.1	Extended: Internet Protocol Security (IPSec) Protocol
User Data Protection	FDP_RIP.2	Full residual information protection
Identification and Authentication	FIA_AFL.1	Authentication Failure Handling
	FIA_PMG_EXT.1	User Identification and Authentication
	FIA_UIA_EXT.1	Extended: Password-based Authentication Mechanism
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_X509_EXT.1	Extended: X.509 Certificates
	FIA_PSK_EXT.1	Extended: Pre-Shared Key Composition
Security Management	FMT_MOF.1	Management of Security Function Behavior
	FMT_MTD.1	Management of TSF Data (for general TSF data)
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Security Roles
Protection of the TSF	FPT_FLS.1	Fail Secure
	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_APW_EXT.1	Extended: Protection of Administrator Passwords

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
	FPT_STM.1	Reliable Time Stamps
	FPT_TUD_EXT.1	Extended: Trusted Update
	FPT_TST_EXT.1	TSF Testing
TOE Access	FTA_SSL_EXT.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/Channels	FTP_ITC.1(1), (2)	Inter-TSF trusted channel
	FTP_TRP.1	Trusted path
Stateful Traffic/Packet Filtering	FFW_RUL_EXT.1	Stateful Traffic Filtering
	FPF_RUL_EXT.1	Packet Filtering

Table 6-1 – TOE Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 *FAU_GEN.1 Audit Data Generation*

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

- Start-up and shut-down of the audit functions;
- All auditable events for the not specified level of audit; and
- All Administrative actions;
- [*Specifically defined auditable events listed in Table 6-2 - Audit Events and Details,* Table 6-3 - Audit Events and Details from FWEP*, and* Table 6-4 - Audit Events and Details from VPNEP].

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 6-2 - Audit Events and Details,* Table 6-3 - Audit Events and Details from FWEP*, and* Table 6-4 - Audit Events and Details from VPNEP].

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FCS_CKM.1(1), (2), (3).	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2), (3	None.	
FCS_COP.1(4)	None.	
FCS_COP.1(5)	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_PSK_EXT.1	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
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	Indication that TSF self-test was completed.	Any additional information generated by the tests beyond “success” or “failure”.
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.	

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FTP_ITC.1(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 6-2 - Audit Events and Details from NDPP

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets

Table 6-3 - Audit Events and Details from FWEP

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FCS_IPSEC_EXT.1	Session Establishment with peer	Source and destination addresses Source and destination ports TOE Interface
FIA_X509_EXT.1	Establishing session with CA	Source and destination addresses Source and destination ports TOE Interface
FIA_PSK_EXT.1	None.	None.
FPT_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets

Table 6-4 - Audit Events and Details from VPNEP

6.1.1.2 *FAU_GEN.2 User Identity Association*

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 *FAU_STG_EXT.1 External Audit Trail Storage*

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 IPSec protocol.

6.1.2 Cryptographic Support

6.1.2.1 *FCS_CKM.1(1) Cryptographic Key Generation (for asymmetric keys)*

FCS_CKM.1.1(1) **Refinement:** 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.

Application Note: This SFR comes from the NDPP.

6.1.2.2 *FCS_CKM.1(2) Cryptographic Key Generation (for asymmetric keys)*

FCS_CKM.1.1(2) **Refinement:** 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 elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, "Digital Signature Standard"
- 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 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes

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

Application Note: This VPNEP requires specific algorithms to be used in key establishment, and this instantiation of the requirement from the NDPP ensures the right selections are made.

6.1.2.3 *FCS_CKM.1 (3) Cryptographic Key Generation (for asymmetric keys)*

FCS_CKM.1.1(3) **Refinement:** The TSF shall generate **asymmetric** cryptographic keys used **for IKE peer authentication** in accordance with **a**:

- FIPS PUB 186-3, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes;
- FIPS PUB 186-3, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and no other curves;

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

Application Note: From the VPNEP, the ANSI X9.31-1998 option will be removed from the selection in a future publication of this document. Presently, the selection is not exclusively limited to the FIPS PUB 186-3 options in order to allow industry some further time to complete the transition to the modern FIPS PUB 186-3 standard.

The keys that are required to be generated by the TOE through this requirement are intended to be used for the authentication of the VPN peers during the IKE (either v1 or v2) key exchange. While it is required that the public key be associated with an identity in an X509v3 certificate, this association is not required to be performed by the TOE, and instead is expected to be performed by a Certificate Authority in the Operational Environment.

As indicated in FCS_IPSEC_EXT.1, the TOE is required to implement support RSA or ECDSA (or both) for peer authentication.

The generated key strength of 2048-bit RSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

6.1.2.4 *FCS_CKM_EXT.4 Cryptographic Key Zeroization*

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

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

FCS_COP.1.1(1) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm *AES operating in* **GCM, CBC**, ***no other***

modes and cryptographic key sizes 128-bits, 256-bits, and **no other key sizes** that meets the following:

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

Application Note: The VPNEP requires the modes GCM and CBC to be used in the IPsec and IKE protocols (FCS_IPSEC_EXT.1.4, FCS_IPSEC_EXT.1.6). Therefore, the FCS_COP.1.1(1) element in the NDPP has been specified here to ensure the ST Author includes these two modes to be consistent with the IPsec requirements.

6.1.2.6 *FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)*

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a **RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater**

that meets the following:

- **RSA Digital Signature Algorithm**
  - **FIPS PUB 186-2, "Digital Signature Standard"**

6.1.2.7 *FCS_COP.1(3) Cryptographic Operation (for cryptographic signature)*

FCS_COP.1.1(3) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a **Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater**

that meets the following:

- **Elliptic Curve Digital Signature Algorithm**
  - **FIPS PUB 186-3, "Digital Signature Standard"**
  - **The TSF shall implement "NIST curves" P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, "Digital Signature Standard").**

6.1.2.8 *FCS_COP.1(4) Cryptographic Operation (for cryptographic hashing)*

FCS_COP.1.1(4) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **SHA-1, SHA-256, SHA-384**

and **message digest sizes 160, 256, 384 bits**. that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

6.1.2.9 *FCS_COP.1(5) Cryptographic Operation (for keyed-hash message authentication)*

FCS_COP.1.1(5) **Refinement:** The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC- **SHA-1, SHA-256**, **key size *160, 256* (in bits) used in HMAC**, **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."

6.1.2.10 *FCS_RBG_(EXT).1 Extended: Cryptographic Operation (Random Bit Generation)*

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 (384) 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 384 bits of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

Application Note: The NDPP allows the ST Author to choose whether the noise source is software based or hardware based. For compliance with the VPNEP, there must be at least one hardware based noise source.

6.1.2.11 *Explicit: SSH (FCS_SSH_EXT)*

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 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-sha1-96.

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange method used for the SSH protocol.

6.1.2.12 *FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (IPsec) Communications*

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The TSF shall implement tunnel mode

Application Note: Future versions of this EP will require that the TSF implement both tunnel mode and transport mode.

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC 4106, AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC.

Application Note: If an AES-CBC selection is made, the SHA-based HMAC must be consistent with what is specified in the NDPP FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication) requirement.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, no other RFCs for extended sequence numbers and RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23) and RFC 4868 for hash functions.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the IKEv1, IKEv2 protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and no other algorithm.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Application Note: Element 1.7 is only applicable if IKEv1 is selected.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that IKEv2 SA lifetimes can be configured by an Administrator based on number of **bytes** ~~packets~~ or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs, IKEv1 SA lifetimes can be configured by an Administrator based on number of **bytes** ~~packets~~ or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.

Application Note: It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting

a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in gx mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least *384 bits that is at least twice the "bits of security" value associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General] bits*.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^*192 "bits of security" value(s) associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General.*

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), *no other DH groups*.

FCS_IPSEC_EXT.1.12 The TSF shall ensure that all IKE protocols perform peer authentication using a RSA, ECDSA that use X.509v3 certificates that conform to RFC 4945 and pre-shared keys.

FCS_IPSEC_EXT.1.13 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection.

6.1.3 User Data Protection (FDP)

6.1.3.1 *FDP_RIP.2 Full Residual Information Protection*

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 *FIA_AFL.1 Authentication Failure Handling*

FIA_AFL.1.1 **Refinement:** The TSF shall detect when **an Administrator configurable positive integer of successive** unsuccessful authentication attempts occur related to **administrators attempting to authenticate remotely**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall *prevent the offending remote administrator from successfully authenticating until an Administrator defined time period has elapsed*.

Application Note: This requirement does not apply to an administrator at the local console, since it does not make sense to lock a local administrator's account in this fashion. This could be addressed by (for example) requiring a separate account for local administrators or having the authentication mechanism implementation distinguish local and remote login attempts. The "action" taken by a local administrator is implementation specific and would be defined in the administrator guidance (for example, lockout reset or password reset). The ST author chooses one of the selections for handling of authentication failures depending on how the TOE has implemented this handler.

6.1.4.2 *FIA_PMG_EXT.1 Password Management*

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 the complete set of standard ASCII characters and control characters;*
2. Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater;

6.1.4.3 *FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism*

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

6.1.4.4 *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;
- *arp 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.

6.1.4.5 *FIA_UAU.7 Protected Authentication Feedback*

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.

6.1.4.6 *FIA_X509_EXT.1 Extended: X.509 Certificates*

FIA_X509_EXT.1.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and no other protocols connections.

FIA_X509_EXT.1.2 The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.1.3 The TSF shall provide the capability for authenticated Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

FIA_X509_EXT.1.4 The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

Application Note: The public key referenced in FIA_X509_EXT.1.4 is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1(2).

FIA_X509_EXT.1.5 The TSF shall validate the certificate using the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759.

Application Note: While the choice of revocation method employed is left to the ST author, future versions of this EP will mandate both methods be available to the TOE's Administrator.

FIA_X509_EXT.1.6 The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.

FIA_X509_EXT.1.7 The TSF shall not treat a certificate as a CA certificate if the basicConstraints extension is not present or the cA flag is not set to TRUE.

FIA_X509_EXT.1.8 The TSF shall not establish an SA if a certificate or certificate path is deemed invalid.

FIA_X509_EXT.1.9 The TSF shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

FIA_X509_EXT.1.10 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall, at the option of the administrator, establish an SA or disallow the establishment of an SA.

Application Note: The intent of FIA_X509_EXT.1.8 is that the TOE is configurable to allow or disallow session establishment if the TOE cannot connect to an entity responsible for providing certificate validation information. For instance, if a CRL cannot be obtained because a machine is down, or the network path is broken, the administrator may elect to configure the TOE to allow sessions to continue to be established, rather than terminate the TOE's ability to establish any new SAs because it cannot reach the CA.

6.1.4.7 *FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition*

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec and no other protocols.

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and *1 to 255 characters*;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using SHA-1, the *TOE converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the pseudo-random function that is configured as the hash algorithm for the IKE exchanges* and be able to use no other pre-shared keys.

FIA_PSK_EXT.1.4 The TSF shall be able to accept bit-based pre-shared keys.

6.1.5 Security Management (FMT)

6.1.5.1 *FMT_MOF.1 Management of Security Functions Behavior*

FMT_MOF.1.1 **Refinement:** The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this EP to an authenticated Administrator.

Application Note: The EP refers to the VPNEP.

6.1.5.2 *FMT_MTD.1 Management of TSF Data (for general TSF data)*

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

6.1.5.3 *FMT_SMF.1 Specification of Management Functions*

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;*
- *Ability to configure Firewall rules (from FWEP). and*
- *Ability to configure the cryptographic functionality,* (from VPNEP)
- *Ability to configure the IPsec functionality,* (from VPNEP)
- *Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator,* (from VPNEP)
- *Ability to configure all security management functions identified in other sections of this EP.* (from VPNEP)
- *No other capabilities.*

6.1.5.4 *FMT_SMR.2 Restrictions on Security Roles*

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

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

6.1.6.2 *FPT_APW_EXT.1Extended: Protection of Administrator Passwords*

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 *FPT_FLS.1 Fail Secure*

FPT_FLS.1.1 **Refinement:** The TSF shall **shutdown** when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

Application Note: The failures relevant to this requirement are the FPT_TST_EXT.1.1 requirement in the NDPP, and the FPT_TST_EXT.1.2 requirement specified in the VPNEP.

6.1.6.4 *FPT_STM.1 Reliable Time Stamps*

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

6.1.6.5 *FPT_TUD_EXT.1 Extended: Trusted Update*

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 digital signature mechanism prior to installing those updates.

Application Note: The NDPP provides an option of which method of verification the ST Author wishes to specify. For compliance with the VPNEP, a digital signature mechanism (one of those specified in FCS_COP.1(2) must be employed.

6.1.6.6 *FPT_TST_EXT.1: TSF Testing*

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.

FPT_TST_EXT.1.2 The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

Application Note: The NDPP contains one element for this component, which simply requires a suite of self-tests to demonstrate correct operation of the TSF. This element is added to that component to comply with the VPNEP.

6.1.7 TOE Access

6.1.7.1 *FTA_SSL_EXT.1 TSF-initiated Session Locking*

FTA_SSL_EXT.1.1 The TSF shall for local interactive sessions,

- terminate the session

after a Security Administrator-specified time interval of session inactivity.

6.1.7.2 *FTA_SSL.3 TSF-initiated Termination*

FTA_SSL.3.1 **Refinement:** The TSF shall terminate a **remote** interactive session after a *[Security Administrator-configurable time interval of session inactivity*].

6.1.7.3 *FTA_SSL.4 User---initiated Termination*

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

6.1.7.4 *FTA_TAB.1 Default TOE Access Banners*

FTA_TAB.1.1 **Refinement:** Before establishing **an administrative user** session the TSF shall display a **Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

6.1.8 Trusted Path/Channel (FTP)

6.1.8.1 *FTP_ITC.1(1) Inter-TSF Trusted Channel (Prevention of Disclosure)*

FTP_ITC.1.1(1) **Refinement:** The TSF shall **use IPSec** 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(1) The TSF shall permit the TSF, or the **authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3(1) The TSF shall initiate communication via the trusted channel for *export of audit logs to syslog servers*.

6.1.8.2 *FTP_ITC.1(2) Inter-TSF trusted channel*

FTP_ITC.1.1(2) **Refinement:** The TSF shall **use IPsec, and no other protocols** to provide a **trusted** communication channel between itself and **all authorized IT entities** 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.

Application Note: The NDPP allows trusted channels other than IPsec to be available for communication with external IT entities. To be compliant with the VPNEP, the selection is made such that the TOE must provide the IPsec protocol as a configurable option to the administrator.

6.1.8.3 *FTP_TRP.1Trusted Path*

FTP_TRP.1.1 **Refinement:** 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 **Refinement:** 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.*

6.1.9 Stateful Traffic/Packet Filtering (FFW and FPF)

6.1.9.1 *FFW_RUL_EXT.1 Stateful Firewall Filtering*

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

FFW_RUL_EXT.1.3 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
    - Type
    - Code
- ICMPv6
    - Type
    - Code
- IPv4
    - Source address
    - Destination Address
    - Transport Layer Protocol
- IPv6

- Source address
- Destination Address
- Transport Layer Protocol

- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface.

FFW_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

FFW_RUL_EXT.1.5 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, ICMP based on the following network packet attributes:

1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
2. UDP: source and destination addresses, source and destination ports;
3. ICMP*:* source and destination addresses, type, code, no other protocols.

b) Remove existing traffic flows from the set of established traffic flows based on the following: session inactivity timeout, completion of the expected information flow.

FFW_RUL_EXT.1.7 The TSF shall be able to process the following network protocols:

1. FTP,

2. no other protocols,

to dynamically define rules or establish sessions allowing network traffic of the following types:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- none.

FFW_RUL_EXT.1.8 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;
3. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
4. The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;

9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4;
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. no other rules.

FFW_RUL_EXT.1.9 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

6.1.9.2 *FPF_RUL_EXT.1 Packet Filtering*

FPF_RUL_EXT.1.1 The TSF shall perform Packet Filtering on network packets processed by the TOE.

FPF_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 791 (IPv4)
- RFC 2460 (IPv6)

- RFC 793 (TCP)
- RFC 768 (UDP)

FPF_RUL_EXT.1.3 The TSF shall allow the definition of Packet Filtering rules using the following network protocol fields:

- IPv4
    - Source address
    - Destination Address
    - Protocol
- IPv6
    - Source address
    - Destination Address
    - (Next Header) Protocol
- TCP
    - Source Port
    - Destination Port
- UDP
    - Source Port
    - Destination Port

and distinct interface.

FPF_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Packet Filtering rules: permit, deny, and log.

FPF_RUL_EXT.1.5 The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.6 The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.5) in the following order: Administrator-defined.

FPF_RUL_EXT.1.7 The TSF shall deny packet flow if a matching rule is not identified.

6.2 CC Component Hierarchies and Dependencies

This section of the ST demonstrates that the identified SFRs include the appropriate hierarchy and dependencies. This ST follows exactly the security requirements included in the NDPP, FWEP, and VPNEP. Any hierarchies and dependencies are satisfied in accordance with the NDPP, FWEP, and VPNEP.

6.3 Security Assurance Requirements

The Security Assurance Requirements for this evaluation are listed in Section 6.4.3 – Security Assurance Requirements.

6.4 Security Requirements Rationale

6.4.1 Security Functional Requirements

This ST follows exactly the NDPP, FWEP, and VPNEP and all of the security functional requirements within. The PPs map SFRs to objectives in Section 3 of the NDPP, Section 4 of the FWEP, and Section 3 in the VPNEP.

6.4.2 Sufficiency of Security Requirements

The following table presents a mapping of the rationale of TOE Security Requirements to Objectives as described in the NDPP Section 3, FWEP Section 4, and VPNEP Section 3.

OBJECTIVE	SFR
Protected Communications O.PROTECTED_COMMUNICATIONS	FCS_CKM.1 (1), (2), (3) FCS_CKM_EXT.4 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_COP.1(5) FCS_RBG_EXT.1 FCS_SSH_EXT.1 FTP_ITC.1(1), (2) FTP_TRP.1
Verifiable Updates O.VERIFIABLE_UPDATES	FPT_TUD_EXT.1 FCS_COP.1(2) FCS_COP.1(4)
System Monitoring O.SYSTEM_MONITORING	FAU_GEN.1 FAU_GEN.2 FAU_STG_EXT.1 FPT_STM.1

OBJECTIVE	SFR
TOE Administration O.TOE_ADMINISTRATION O.DISPLAY_BANNER O.SESSION_LOCK	FIA_UIA_EXT.1 FIA_PMG_EXT.1 FIA_UAU_EXT.2 FIA_UAU.7 FMT_MTD.1 FMT_SMF.1 FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4
Residual Information Clearing O.RESIDUAL_INFORMATION_CLEARING	FDP_RIP.2
TSF Self Test O.TSF_SELF_TEST	FPT_TST_EXT.1

Table 6-5 – Rationale for TOE SFRs to Objectives from NDPP

OBJECTIVE	SFR
O.ADDRESS_FILTERING	FFW_RUL_EXT.1
O.PORT_FILTERING	FFW_RUL_EXT.1
O.STATEFUL_INSPECTION	FFW_RUL_EXT.1
O.RELATED_CONNECTION_FILTERING	FFW_RUL_EXT.1
O.SYSTEM_MONITORING	FAU_GEN.1 FFW_RUL_EXT.1
O.TOE_ADMINISTRATION	FMT_SMF.1

Table 6-6 – Rationale for TOE SFRs to Objectives from FWEP

OBJECTIVE	SFR
O.ADDRESS_FILTERING	FPF_RUL_EXT.1
O.CRYPTOGRAPHIC_FUNCTIONS	FCS_CKM.1(1), (2), (3) FCS_COP.1(1) FCS_RBG_EXT.1 FCS_IPSEC_EXT.1
O.AUTHENTICATION	FTP_ITC.1(1), (2) FCS_IPSEC_EXT.1 FIA_PSK_EXT.1
O.FAIL_SECURE	FPT_FLS.1
O.TOE_ADMINISTRATION	FMT_SMF.1 FIA_AFL.1

Table 6-7 - Rationale for TOE SFRs to Objectives from VPNEP

6.4.3 Security Assurance Requirements

The assurance security requirements for this Security Target are from the NDPP, FWEP, and VPNEP. The assurance components are summarized in the following table:

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
ADV: Development	ADV_FSP.1	Basic Functional Specification
AGD: Guidance Documents	AGD_OPE.1	Operational User Guidance

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
	AGD_PRE.1	Preparative Procedures
ALC: Lifecycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
ATE: Tests	ATE_IND.1	Independent Testing - Sample
AVA: Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 6-8 – Security Assurance Requirements

Detailed assurance activities are described in NDPP Section 4.2, FWEP Section 4.2 and 4.3, and VPNEP Section 5.

6.4.4 Security Assurance Requirements Rationale

The ST specifies assurance activities specified in the NDPP, FWEP, and VPNEP.

6.4.5 Security Assurance Requirements Evidence

This section identifies the measures applied to satisfy CC assurance requirements.

SECURITY ASSURANCE REQUIREMENT	EVIDENCE TITLE
ADV_FSP.1 Basic functional specification	Functional Specification: Juniper Networks, Inc. Junos 12.1 X46 D20 for SRX and LN Series Platforms
AGD_OPE.1 Operational User Guidance AGD_PRE.1 Preparative Procedures	Operational User Guidance and Preparative Procedures Supplement: Juniper Networks, Inc. Junos 12.1 X46 D20 for SRX and LN Series Platforms
ALC_CMC.1 Labeling of the TOE ALC_CMS.1 TOE CM Coverage	Configuration Management Processes and Procedures: Juniper Networks, Inc. Junos 12.1 X46 D20 for SRX and LN Series Platforms
ATE_IND.1 Independent Testing	Provided by Evaluation Lab
AVA_VAN.1 Vulnerability Analysis	Vulnerability Analysis: Juniper Networks, Inc. Junos 12.1 X46 D20 for SRX and LN Series Platforms

Table 6-9 – Security Assurance Rationale and Measures

7 TOE Summary Specification

This section presents the Security Functions implemented by the TOE.

7.1 TOE Security Functions

The security functions performed by the TOE are as follows:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channel
- Stateful Firewall/Packet Filtering (FWEP & VPNEP)

7.2 Security Audit

JUNOS creates and stores audit records which contain the date and time of the event, type of event, subject (user) identity and the outcome of the event for the following events (the detail of content recorded for each audit event is detailed in Table 6-2 and 6-3): :

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.
m) Application of firewall rules configured with the 'log' operation

The TOE records the following with each log entry:

- Date and time of the event,
- type of event,
- subject identity,

- the outcome (success or failure) of the event;

Auditing is done using syslog. The log entries are stored in the order the event was recorded which preserves the temporal order of the events. Syslog can be configured to store the audit logs locally, or to send them to one or more syslog log servers (via IPSec). Local audit log are stored in /var/log/. Only an authorized administrator can read log files, or delete log and archive files. 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 *Junos OS System Basics Configuration Guide* 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* Chapter 13, Subsection "archive (All System Log Files)".

The maximum value that can be specified for the size of a log file is 1GB. These defaults maximum sizes can be modified by the user, as detailed in *Junos OS System Basics Configuration Guide* Chapter 9, Subsection 'Specifying Log File Size, Number, and Archiving Properties'.
For more information about configuring event logging, see the *Junos OS System Basics Configuration Guide* and the *Junos OS Secure Configuration Guide for Common Criteria Network Device Protection Profile for Devices Running Junos OS 12.1.*

The TOE is configured to direct syslog traffic from the TOE to an external syslog server via a IPSec VPN tunnel. There is no local audit log storage. A VPN tunnel is initiated from the TOE immediately upon configuration commit and communications from TOE to the syslog server is encrypted. Audit records are sent to the syslog server periodically as configured by the Administrator. Maximum log sizes are configurable by the Administrator.By default, the TOE stores all logs locally. The level of what to log locally is configurable. The TOE is configurable to forward copies of local logs to an external syslog server. Similar to local storage, the TOE can be configured to determine what local logs to forward to external syslog server

When the TOE is set up for an external syslog server, will there be any local storage of audit events The TOE forwards copies of local logs to the external syslog server and retains local copies of all logs when the TOE is configured with event log mode. In stream log mode, all logs except traffic log are stored locally and can be also forwarded to external syslog server, whereas traffic logs are not stored locally but only forwarded to external syslog server

The connection between the TOE and syslog server is established on an event basis depending on pre-configuration of what type of logs to be forwarded from local to external. When the configured condition is met, the TOE sends local logs to external syslog server.

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

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

7.3 Cryptographic Support

All cryptographic functions implemented by the secure network appliance are implemented in the JUNOS crypto module. The TOE meets cryptographic requirements either by allowing the administrator to enable the appropriate cryptographic functions. The Cryptographic security function is described in the context of how it satisfies the cryptographic security requirements.

The cryptographic algorithms have been tested and issued the following certificate numbers by the Cryptographic Algorithm Validation Program:

CRYPTOGRAPHIC ALGORITHM	OPERATIONAL ENVIRONMENT	CAVP CERTIFICATE NUMBER
AES-GCM	Junos FIPS Version 12.1X46.D20 – Data Plane	Val# 3353
AES-CBC	Junos FIPS Version 12.1X46.D20 – Data Plane	Val# 3353
AES-CBC	Junos FIPS Version 12.1X46.D20 – Authentec	Val# 3402
AES-CBC	Junos FIPS Version 12.1X46.D20 – OpenSSL	Val# 3354
SHA	Junos FIPS Version 12.1X46.D20 – Data Plane	Val# 2779
SHA	Junos FIPS Version 12.1X46.D20 – Authentec	Val# 2815
SHA	Junos FIPS Version 12.1X46.D20 – OpenSSL	Val# 2780
HMAC-SHA	Junos FIPS Version 12.1X46.D20 – Data Plane	Val# 2135
HMAC-SHA	Junos FIPS Version 12.1X46.D20 – Authentec	Val# 2170
HMAC-SHA	Junos FIPS Version 12.1X46.D20 – OpenSSL	Val# 2136
HMAC_DRBG	Junos FIPS Version 12.1X46.D20 – OpenSSL	Val# 785
ECDSA	Junos FIPS Version 12.1X46.D20 –Authentec	Val# 677
ECDSA	Junos FIPS Version 12.1X46.D20 – Authentec KeyGen	Val# 669
ECDSA	Junos FIPS Version 12.1X46.D20 – OpenSSL	Val# 665
ECDSA	Junos FIPS Version 12.1X46.D20 – OpenSSL KeyGen	Val #687
DSA	Junos FIPS Version 12.1X46.D20 – Authentec KeyGen	Val# 958
DSA	Junos FIPS Version 12.1X46.D20 – OpenSSL KeyGen	Val# 965
RSA	Junos FIPS Version 12.1X46.D20 – Authentec	Val# 1741
RSA	Junos FIPS Version 12.1X46.D20 – OpenSSL KeyGen	Val# 1719

Table 7-1 – CAVS Certificate Results

The JUNOS crypto-module 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. Asymmetric keys are generated (FCS_CKM.1(1)) in accordance with NIST SP 800-56A for use with SSH to the admin console. Asymmetric keys are generated (FCS_CKM.1(2)) in accordance with NIST SP 800-56A and FIPS PUB 186-3 for IPsec key establishment. TOE implements all of the "shall" and "should" requirements and none of the "shall not" or "should not" from FIPS PUB 186-3 Appendix B3

and B4. Asymmetric keys are generated (FCS_CKM.1(3)) in accordance with FIPS PUB 186-3 for IKE used with IPsec.

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 SSH_ECDSA authentication methods and password-based authentication for SSHv2.

Packets greater than 256 kb in an SSH transport connection are dropped and the connection is terminated by the TOE. The SSH daemon maintains a 256K buffer for incoming packet processing, adding to the buffer in 1K increments. If the accumulated data for a packet exceeds the buffer size, the packet is dropped, the accumulator buffer is reset to zero and a log message indicating that the packet was dropped is created.

The TOE supports AES-CBC-128and AES-CBC-256 encryption algorithms for SSH transport and uses SSH-ECDSA as its public key algorithm.

The data integrity algorithms used in SSH transport connection are "hmac-sha1" as required by [RFC4253]

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

The TOE supports keyed hash message authentication using HMAC-SHA-that meet FIPS Pub 198-1. The TOE supports cryptographic hashing via the SHA-1 algorithms provided it has a message digest size of 160 bits.

The TOE handles zeroization for all CSP, plaintext secret and private cryptographic keys according to the table 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].

The TOE sets a limit on packets at 256Kb. When a packet exceeds that limit, the packet is dropped, the buffer discarded and a log entry indicating that a packet with bad length was received.

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. Keys are zeroized with the Zeroize Command.
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. Keys are zeroized at reboot time.
User Password	Plaintext value as entered by user	Hashed	Memory	Overwritten with zero's with Delete command.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.
SSH Host public key	1st time SSH is configured the key is generated. RSA(>=1k), DSA. Idenitfy the host.	Plaintext	Memory	Public keys stored in memory are zeroized when the memory is no longer needed. Public values stored in flash are not zeroized unless an explicit "request system zeroize" is executed
SSH client keys	Session keys used with SSH	Plaintext	Memory	Reboot
DH Private Exponent	Used to authenticate users to the module.	Plaintext	Memory	Clear IKE SA command or reboot the box
DH Public Key	Ephemeral Diffie-Hellman public key used in SSH and IKE key establishment.	Plaintext	Memory	Public keys stored in memory are zeroized when the memory is no longer needed. Public values stored in flash are not zeroized unless an explicit "request system zeroize" is executed

Table 7-2 - Key Zeroization Handling

Public keys stored in memory are zeroized when the memory is no longer needed. Public values stored in flash are not zeroized unless an explicit "request system zeroize" is executed.

See also Section 7.12 800-56 Conformance Statements for descriptions of RFC conformance.

7.3.1 IPSEC Support

The TOE is conformant to RFC 4301. (FCS_IPSEC_EXT.1.1)

The TOE supports tunnel mode only. (FCS_IPSEC_EXT.1.2)

By default, the TOE denies all traffic through an SRX Series device. In fact, an implicit default security policy exists that denies all packets. You can change this behavior by configuring a standard security policy that permits certain types of traffic. The implicit default policy can be changed to permit all traffic with the '`set security policies default-policy`' command; however, this is *not* recommended.

The security policy rule set is an ordered list of security policy entries, each of which contains the specification of a network flow and an action:

- Source IP address and network mask
- Destination IP address and network mask
- Protocol
- Source port
- Destination port
- Action: permit, deny, drop silently, log

Each packet is compared against entries in the security policy ruleset in sequential order until one is found that matches the specification in the policy, or until the end of the rule set is reached, in which case the implicit default policy is implemented. (FCS_IPSEC_EXT.1.3)

The TOE supports AES-GCM-128 and AES-GCM-256, and AES-CBC-128 or AES-CBC-256 using HMAC SHA-1 and SHA-256. Hashing algorithms including "hmac-sha1-96","hmac-sha-256-128" can be configured for AES-CBC. AES-GCM does not permit a hashing algorithm to be configured, as it performs its own hashing. (FCS_IPSEC_EXT.1.4)

IKEv1 and IKEv2 are implemented. IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, no other RFCs for extended sequence numbers, and RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23) and RFC 4868 for hash functions. (FCS_IPSEC_EXT.1.5)

The TOE supports only CBC mode is supported for the encrypted IKEv1 and IKEv2 payloads using AES-CBC-128, AES-CBC-256. (FCS_IPSEC_EXT.1.6)

Main mode is only used for IKEv1 Phase 1 exchanges. In the evaluated configuration, the TOE permits only main mode to be configured. There is no option to configure aggressive mode. The CLI used is set `security ike policy <name> mode main`. (FCS_IPSEC_EXT.1.7)

The following CLI commands configure a lifetime of either kilobytes or seconds: (FCS_IPSEC_EXT.1.8)

```
set security ipsec proposal <name> lifetime-kilobytes <kb>
```

```
set security ipsec proposal <name> lifetime-seconds <seconds>
```

The TOE by default uses HMAC DRBG with SHA-384 to provide 192 bits of strength. (FCS_IPSEC_EXT.1.9, FCS_IPSEC_EXT.1.10)

The TOE supports Diffie-Hellman Groups 14, 19, 20, and 24. (FCS_IPSEC_EXT.1.11)

The TOE supports both RSA and ECDSA (FCS_IPSEC_EXT.1.12) for use with X.509v3 certificates that conform to RFC 4945 and pre-shared Keys for IPsec support.

The TOE ensures that the strength of the symmetric algorithm (128 or 256 bits) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection. (FCS_IPSEC_EXT.1.13)

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

- FCS_CKM.1(1), (2), (3)
- FCS_CKM_EXT.4
- FCS_COP.1(1)
- FCS_COP.1(2)
- FCS_COP.1(3)
- FCS_COP.1(4)
- FCS_COP.1(5)
- FCS_RBG_(EXT).1
- FCS_SSH_EXT.1
- FCS_IPSEC_EXT.1

7.4 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. 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

7.5 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. The password has a minimum length of 15[1] 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 TOE provides obscured feedback during the authentication process.

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 (ie '~/.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

[1] 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

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()[2] 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 keys) 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.

A remote administrator may logon to the TOE via SSH. Administrator credentials are stored locally to the device. If the remote administrator presents a valid username and password, access to the TOE is granted. If the credentials are invalid, the TOE allows the authentication to be retried after an interval that starts at 1 second at increases exponentially. If the number of authentication attempts exceeds the configured maximum, no authentication attempts are accepted for a configured time interval. When the interval has expires, authentication attempts are again accepted. (FIA_AFL.1)

Certificates are stored in non-volatile flash memory. Access to flash memory requires administrator credentials. A certificate may be loaded via command line. Alternately, the key-pair may be generated locally and the certificate self-signed. (FIA_X.509_EXT.1)

The TOE uses pre-shared keys for IPSec and no other protocols The TOE accepts ASCII pre-shared or bit-based keys of 1 to 255 characters (and their binary equivalent) that may contain upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. The TOE accepts pre-shared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges. (FIA_PSK_EXT.1).

The TOE uses X.509 certificates as defined in RFC 5280.

To generate a Certificate Request, t*he administrator uses the “request security pki generate-certificate-request” CLI command and supplies the following values:*

- Certificate-id – The internal identifier string for this certificate
- Domain-name
- Email address
- IP address
- Subject (DC=<Domain component>,CN=<Common-Name>,OU=<Organizational-Unit-name>,O=<Organization-name>,SN=<Serial-Number>,L=<Locality>,ST=<state>,C=<Country>)
- Filename – The local file in which to store the certificate signing request

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

To validate certificates, the TOE extracts the subject, issuer, subjects public key, signature, basicConstraints and validity period fields. If any of those fields is not present, the validation fails. The issuer is looked up in the PKI database. If the issuer is not present, or if the issuer certificate does not have the CA:true flag in the basicConstraints section, the validation fails. The TOE verifies the validity of the signature. If the signature is not valid, the validation fails. It then confirms that the current date and time is within the valid time period specified in the certificate.

If the TOE has been configured to perform a revocation check, it may use a CRL or OCSP, but not both simultaneously. If OCSP is selected, it may be configured to use a CRL in the event of a connection failure to the OCSP server.

If the TOE has been configured for CRL revocation checking and the certificate considered for validation is not present on the CRL, then the validation succeeds. If the CRL fails to download, the certificate is considered to have failed validation, unless the option to skip CRL checking on download failure has been enabled.

If the TOE has been configured for OCSP, and the response from the OCSP responder is “good” or “unknown”, then the validation succeeds. If there is an error, or no response from the OCSP responder, then the TOE will either fail the validation, skip the OCSP check and pass the validation, or fall back to CRL checking, as configured.

The TOE validates a certificate path by building a chain of certificates based upon issuer and subject linkage, validating each according the certificate validation procedure described above. If any certificate in the chain fails validation, the validation fails as a whole. A self-signed certificate is not required to be at the root of the certificate chain.

The TOE determines if a certificate is a CA certificate by requiring the CA:true flag to be present in the basicConstraints section.

The TOE requires that the configured IKE identity of the local and remote endpoints to match the contents of the certificate associated with a SA endpoint. The TOE permits the identity to be expressed as distinguished name, email address, fully qualified domain name or IP address. If either certificate does not validate, or the contents do not match the configured identity, then the SA will not be established.

If the TSF cannot establish a connection to determine the validity of a certificate, by default the SA will not be established.

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

- FIA_AFL.1
- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.2

- FIA_UAU.7
- FIA_X509_EXT.1
- FIA_PSK_EXT.1

7.6 Security Management

There is only one user role defined for the TOE: Authorized Administrator. Because only Authorized Administrator users can be defined on the TOE, non-administrator users can have no access to TOE management functions. 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 (ie '~/.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 keys) before any access to the system is granted. Prior to authentication, the Authorized Administrator only has access to the login screen. 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 (ie '~/.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[3]
- 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;
- Ability to configure Firewall rules;
- Ability to configure the VPN-associated cryptographic functionality;
- Ability to configure the IPsec functionality,
- Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this VPNEP;
- Ability to configure all other VPN-associated security management functions.)

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

Detailed topics on the secure management of Juniper's routers & switches are discussed in the *Junos OS System Basics Configuration Guide* and the *Junos OS Secure Configuration Guide for Common Criteria Network Device Protection Profile for Devices Running Junos OS 12.1*.

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

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

7.7 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. The system clock is also used to determine SA lifetimes and authentication failure timeout.

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 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.

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.

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

Juniper Networks routing platforms run only binaries supplied by Juniper Networks. Each JUNOS software image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. JUNOS software will not execute any binary without a registered signature. This feature protects the system against unauthorized software and activity that might compromise the integrity of your router. These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE.

JUNOS-FIPS and JUNOS 7.5+ use verifi-exec digital signatures (veriexec from NetBSD) to allow the kernel to only execute binaries for which it has a matching SHA1 fingerprint manifests. In JUNOS these fingerprints are loaded from a digitally signed manifest, and the loader will only do so if it can verify the signature. JUNOS uses a standard RSA encrypted SHA1 digest for its signatures.

The power on self-tests may produce some or all of the output shown in the figure below:

```
Starting Memory POST...
Checking datalines... OK
Checking address lines... OK
Checking 512K memory for U-Boot... OK.
Running U-Boot CRC Test... OK.
Flash:  4 MB
USB:   scanning bus for devices...
Root Hub 0: 4 USB Device(s) found
Root Hub 1: 1 USB Device(s) found
    scanning bus for storage devices... 2 Storage Device(s) found
Clearing DRAM..... done
BIST check passed.
1:00:00.0 Vendor/Device ID = 0x811210b5
1:01:07.0 Vendor/Device ID = 0xc72414e4
Net:   octeth0
POST Passed
```

Figure 2 - Self Test Example

In the event of a transiently corrupt state or failure condition, the system will panic; 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 Secure Configuration Guide for Common Criteria Network Device Protection Profile for Devices Running Junos OS 12.1*.

A registered user may download software from support.juniper.net. (Login credentials are required.) For each release, SHA1 hash values are listed on the site. After downloading the software, the user runs a hash utility on the downloaded file and compares the output of the utility with the hash checksum listed on the Juniper download site.

In addition, the installable firmware package has a digital signature that is checked when the administrator attempts to install the package. The public key installed on the device when it is shipped from the factory is used to verify the signature ion the installable package. If signature verification fails, an error message is displayed and the package is not installed

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. 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.

When any self-test fails, the device halts in an error state. No command line input nor traffic to any interface is processed. The device must be power cycled to attempt to return to operation. (FPT_FLS.1)

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

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

7.8 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.

Authorized Administrators have local and remote access capabilities.

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
- FTA_SSL.3
- FTA_SSL.4
- FTA_TAB.1

7.9 Trusted Path/Channels

The TOE supports and enforces Trusted Channels that protect the communications between the TOE and remote audit server from unauthorized disclosure or modification using SSH. 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 cryptographic module.

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

- FTP_ITC.1(1), (2)
- FTP_TRP.1

7.10 Stateful Traffic/Packet Filtering (FWEP and VPNEP)

The boot sequence of the TOE appliances also aids in establishing the securing domain and preventing tamping or bypass of security functionality. The following steps list the boot sequence for the TOE:

- BIOS hardware and memory checks
- Loading and initialization of the FreeBSD Kernel OS
- FIPS self-tests and firmware integrity tests are executed

- The init utility is started (mounts file systems, sets up network cards to communicate on the network, and generally starts all the processes that usually are run on a FreeBSD system at startup)
- Daemon programs such as Internet Service Daemon (INETD), Routing Protocol Daemon (RPD), Syslogd are started; Routing and forwarding tables are initialized
- Management Daemon (or MGD) is loaded, allowing access to management interface
- Physical interfaces are active

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of kernel and Information Flow subsystems, and these interfaces cannot send or receive packets unless previously configured by an authorized administrator. Since the Management Daemon is not loaded until after the kernel and INETD are initialized, no modification to the security attributes can be made by a user or process other than via the management process. INETD is used to start SSHD which provides secure communication path for remote administrators to manage the TOE.

The trusted and untrusted network connection interfaces on the security appliance are not enabled until all of the components on the appliance are fully initialized; power-up tests are successful and ready to enforce the configured security policies. In this manner, the TOE ensures that administrators are appropriately authorized when they exercise management commands and any network traffic is always subject to the configured information flow policies.

The TOE is configured to associate network interfaces to IP subnets. Source IP addresses are then associated with the network interface.

JUNOS is composed of a number of separate executables, or daemons. If a failure occurs in the “flow” daemon causing it to halt, no packet processing will occur and no packets will be forwarded. A failure in another daemon will not prevent the flow daemon from enforcing the policy rule set.

The Information Flow subsystem is responsible for processing the arriving packets from the network to the TOE's network interface. Based on Administrator-configured policy, interface and zone information, the packet flows through the various modules of the Information Flow subsystem. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces. By default, the TOE behavior is to deny packets when there is no rule match unless another required condition allows the network traffic If a security risk is found in the packet. e.g. denial-of-service attacks, the packet is dropped and an event is logged. The packet does not continue to the next module for processing. If the packet is not dropped by a given module, the interrupt handling routine calls the function for the next relevant module.

The Information Flow subsystem consists of the following modules:

- IP Classification Module
- Attack Detection Module

- Session Lookup Module
- Security Policy Module
- Session Setup Module
- Inetd Module
- Rdp Module

The IP Classification module retrieves information from packets received on the network interface device, classifies packets into several categories, saves classification information in packet processing context, and provides other modules with that information for assisting further processing.

The Attack Detection module provides inline attack detection such as IP Spoofing for the security appliance. This module monitors arriving traffic, performs predefined attack detection services (prevents attacks), and issues actions when an attack is found.

The Session Lookup module performs lookups in the session table which is used for all interfaces based on the information in incoming packets. Specifically, the lookup is based on the exact match of source IP address and port, destination IP address and port, protocol attributes (e.g., SYN, ACK, RST, and FIN), and egress/ingress zone. The input is passed to the module as a set of parameters from the Attack Detection module via a function call. The module returns matching wing if a match is found and 0 otherwise. Sessions are removed when terminated.

The Session Setup module is only available for packets that do not match current established sessions. It is activated after the Session Lookup module. If packet has a matched session, it will skip the session setup module and proceed to the Security Policy module, and other modules. Eventually if the packet is not destined for the TOE, the Network interface will pass the traffic out of the appliance.

The Security Policy module examines traffic passing through the TOE (via Session Setup module) and determines if the traffic can pass based on administrator-configured access policies. The Security Policy module is the core of the firewall and IDP functionalities in the TOE: It is the policy enforcement engine that fulfills the security requirements for the user. The Security Policy module will deny packets when there is no policy match unless another policy allows the traffic.

The Session Setup module performs the auditing of denied packets. If there is a policy to specifically deny traffic, traffic matching this deny policy is dropped and logged in traffic log. If there is no policy to deny traffic, traffic that does not match any policy is dropped and not logged. In either case, Session Setup module does not create any sessions for denied traffic. Sessions are created for allowed traffic.

The INETD module provides internet services for the TOE. The module listens on designated ports used by internet services such as FTP. When a TCP or UDP packet arrives with a particular destination port number, inetd launches the appropriate server program (e.g., SSHD) to handle the connection.

The RPD (Routing Protocol Daemon) module provides the implementations and algorithms for the routing protocols and route calculations. The primary goal of the RPD is to create and maintain the Routing Information Base (RIB), which is a database of routing entries. Each routing entry consists of a

destination address and some form of next hop information. RPD module maintains the internal routing table and properly distributes routes from the routing table to Kernel subsystem used for traffic forwarding at the Network interface.

The TOE performs stateful network traffic filtering on network packets using the following network traffic protocols and network fields conforming to the described RFCs:

PROTOCOL/RFC	FIELDS
Internet Control Message Protocol version 4 (ICMPv4) • RFC 792 (ICMPv4)	• Type • Code
Internet Control Message Protocol version 6 (ICMPv6) • RFC 4443 (ICMPv6)	• Type • Code
Internet Protocol (IPv4) • RFC 791 (IPv4)	• Source address • Destination Address • Transport Layer Protocol
Internet Protocol version 6 (IPv6) • RFC 2460 (IPv6)	• Source address • Destination Address • Transport Layer Protocol
Transmission Control Protocol (TCP) • RFC 793 (TCP)	• Source port • Destination port
User Datagram Protocol (UDP) • RFC 768 (UDP)	• Source port • Destination port

Table 7-3 - Traffic filtering RFCs

Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team.

The TOE shall allow permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.

The TOE accepts network packets if it matches an established TCP, UDP or ICMP session using:

- TCP: source and destination addresses, source and destination ports, sequence number, flags
- UDP: source and destination addresses, source and destination ports
- ICMP: source and destination addresses, type, code, and list of matching attributes

The TOE will remove existing traffic flows due to session inactivity timeout, or completion of the session.

The TOE supports FTP (RFC 959) to dynamically establish sessions allowing network traffic according to Authorized Administrator rules. Session events will be logged in accordance with 'log' operations defined in the rules. Source and destination addresses, source and destination ports, transport layer protocol, and TOE Interface are recorded in each log record.

JUNOS implements what is referred to as an Application Layer gateway (ALG) that inspects FTP traffic to determine the port number used for data sessions. The ALG permits data traffic for the duration of the session, closing the port when the session ends. In this context, "session" refers to the TCP data transfer

connection, not the duration of the FTP control session. JUNOS implements ALGs for a number of protocols.

The TSF shall enforce the following default reject rules with logging on all network traffic:

- invalid fragments;
- fragmented IP packets which cannot be re-assembled completely;
- where the source address is equal to the address of the network interface where the network packet was received;
- where the source address does not belong to the networks associated with the network interface where the network packet was received;
- where the source address is defined as being on a broadcast network;
- where the source address is defined as being on a multicast network;
- where the source address is defined as being a loopback address;
- where the source address is a multicast;
- packets where the source or destination address is a link-local address;
- where the source or destination address is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4;
- where the source or destination address is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6;
- with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;

Packets are checked for validity. “Invalid fragments” are those that violate these rules:

- No overlap
- The total fragments in one packet should not be more than 62 pieces
- The total length of merged fragments should not larger than 64k
- All fragments in one packet should arrive in 2 seconds
- The total queued fragments has limitation, depending on the platform
- The total number of concurrent fragment processing for different packet has limitations depending on platform

The Stateful Traffic Filtering function is designed to satisfy the following security functional requirements:

- FFW_RUL_EXT.1
- FPF_RUL_EXT.1

7.11 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 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 SSHv2 message 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: 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 password 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. 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	RFC synopsis	TOE Handling of Security-Related Protocol Options
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 7-4 – RFC Conformance Statements

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

7.12 800-56 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.

7.12.1 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.

800-56A section	800-56A sub section	Compliance
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	Complies with all "shall" statements. Both static and ephemeral keys 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 and ephemeral public keys used.
	5.6.1.2 ECC Key Pair Generation	Complies with all "shall" statements. Both static and ephemeral keys 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
	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.

800-56A section	800-56A sub section	Compliance
	5.6.2.4 FFC Full Public Key Validation Routine	Comply with "shall" statement.
	5.6.2.5 ECC Full Public Key Validation Routine	Performs full validation.
	5.6.2.6 ECC Partial Public Key Validation Routine	Performs full, not partial validation.
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.
	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.

800-56A section	800-56A sub section	Compliance
	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 7-5 – 800-56A Conformance Statements