Security Target

Junos 12.3 X48-D30 for SRX XLR Platforms (NDPP, TFFWEP, IPSEP)

Document Reference:	Juniper_ST_1.2
Document Status:	Released
Document Version:	1.2
Issue Date:	09 February 2017

Prepared For:

Juniper Networks, Inc.

1194 North Mathilda Avenue

Sunnyvale, CA 94089, USA

www.juniper.net

Prepared By:

BAE Systems Applied Intelligence, Pty Ltd

Level 1, 14 Childers Street

Canberra ACT 2601, Australia

www.baesystems.com/ai

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Juniper Networks, Inc. Junos 12.3 X48-D30 for SRX XLR 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.

Amendment history

Version	Date	Revisions
0.1	13-AUG-15	Initial draft
0.2	29-SEP-16	Updated base on evaluators observation report
0.3	20-OCT-16	Updated to resolve formatting, typos and document convention errors.
1.0	12-Dec-16	Updated in response to draft ETR comments
1.1	17-Jan-17	Updated in response to AAR comments
1.2	09-Feb-17	Updated in response to NIAP comments.

Copyright statement

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 12.3 X48-D30 for SRX XLR Platforms
ST Revision	1.2
ST Publication Date	09 February 2017
Author	BAE Systems Applied Intelligence, Pty Ltd

1.2 TOE Reference

TOE Reference	Juniper Networks, Inc. Junos 12.3 X48-D30 for SRX XLR 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
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	Critical security parameter
DFA	Deterministic Finite Automaton
DES	Data Encryption Standard
DH	Diffie Hellman
DMZ	Demilitarized Zone
DoD	Department of Defense
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
FFC	Finite Field Cryptography
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

IDP	Intrusion Detection and Prevention
IPS	Intrusion Prevention 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

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

Table 1-2 – Acronyms Used in Security Target

1.6 TOE Overview

The TOE is Juniper Networks, Inc. Junos 12.3 X48-D30 for SRX XLR 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 TOE also implements Intrusion Prevention System functionality. It is capable to monitor information flows to detect potential attacks based on both pre-defined attack signature and anomaly characteristics in the traffic.

The Junos 12.3 X48-D30 for SRX XLR Platforms may also be referred to as the TOE in this document. The XLR delimiter in the TOE reference was chosen to differentiate this evaluation from Junos 12.3 X48-D30 for SRX Platforms. XLR refers to the specific range of Broadcom processors implemented in this evaluations hardware platforms.

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.

The TOE supports intrusion detection and prevention functionality, which allows it to detect and react to potential attacks in real time. The detection component of the IPS can be based on attack signatures which specify the characteristics of the potentially malicious traffic based on a variety of packet headers payload data attributes. Anomaly detection based on deviation of the monitored traffic from expected values is also supported.

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.

The TOE is managed and configured via Command Line Interface (CLI), which can be access via a console port or remotely using SSH connections, and does not depend on FTP or SSL to operate correctly.

1.7.2 Physical Boundary

The TOE is a combined hardware/software TOE and is defined as the Junos 12.3 X48-D30 for SRX XLR Platforms. The TOE boundary is shown below.

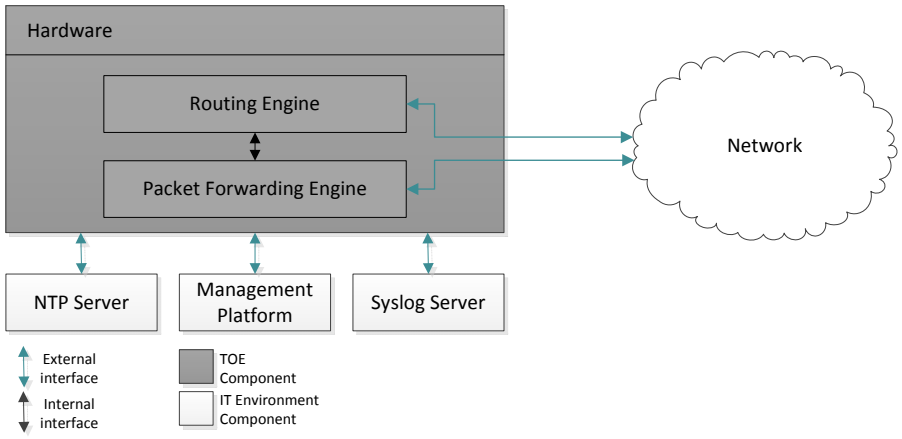

Figure 1 - TOE Boundary

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 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 Version 12.3 X48-D30
Hardware Platforms	SRX1400, SRX3400 and SRX3600; SRX5400, SRX5400E, SRX5600, SRX5600E, SRX5800 and SRX5800E with SPC-2-10-20

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.

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.
Intrusion Prevention	The TOE can be configured to analyze IP-based network traffic forwarded to the TOE's interfaces, and detect violations of administratively-defined IPS policies. The TOE is capable of initiating a proactive response to terminate/interrupt an active potential threat, and to initiate a response in real time that would cause interruption of the suspicious traffic flow.

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)
- TLS

1.7.5 TOE Security Functional Policies

Since the NDPP, FWEP and IPSEP 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 Common Criteria and Junos Evaluated Configuration Guide for SRX Series Security Devices, Release 12.3X48-D30, 20-Jul-16

Junos® OS CLI User Guide, Release 12.3, 13-Jun-16

Junos® OS Installation and Upgrade Guide, Release 12.3, 17-Jun-16

Junos® OS System Basics: Getting Started Configuration Guide, Release 12.3, 10-Jun-16

Junos® OS Intrusion Protection and Prevention Feature Guide for Security Devices, Release 12.3X48-D10, 12-Jan-16

Junos 12.3 X48 for SRX Series Platforms – SRX Guidance Annex, Version 1.0, 17-Jan-17

Junos® OS VPN Feature Guide for Security Devices, Release 12.3X48-D10, 08-07-2016

Junos 12.3 X48 for SRX Series Platforms – SRX Running Processes, Version 1.0, 17-Jan-17

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 #3, 3 November 2014
- 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 for Intrusion Prevention Systems, Version 1.0, 26 June 2014 (IPSEP)

2.2.1 TOE Type Consistency

The NDPP, extended as indicated above, and the TOE describe network device systems, of the following types: firewall and intrusion prevention system.

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, as updated in Security Requirements for Network Devices Errata #3, 3 November 2014, the Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, Version 1.0, 19 December 2011 (FWEP), the Network Device Protection Profile (NDPP) Extended Package for Intrusion Prevention Systems, Version 1.0, 26 June 2014 (IPSEP) 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 [IPSEP].

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 [IPSEP] 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.
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

Note that no additional threats are included in [IPSEP].

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 3-3 – Organizational Security Policy required by NDPP

In addition, the TOE is required to meet the following organizational security policy, as specified in Table 7-3 of [IPSEP]:

POLICY	DESCRIPTION
P.ANALYZ	Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

Table 3-4 – Organizational Security Policy required by IPSEP

3.3 Assumptions

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

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

Table 3-5 – Assumptions from the NDPP

The following assumption regarding the security environment and the intended usage of the TOE, is specified in section 5.1.1 of [FWEP] and Section 7.1.1 of [IPSEP].

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 and IPSEP

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 7.2.1 of [IPSEP].

OBJECTIVES	DESCRIPTION
O.IPSSENSE	The IPS must collect and store information about all events that may indicate an IPS policy violation related to misuse, inappropriate access, or malicious activity on monitored networks.
O.IPSANALYZE	The IPS must apply analytical processes to network traffic data collected from monitored networks and derive conclusions about potential intrusions or network traffic policy violations.
O.IPSREACT	The IPS must respond appropriately to its analytical conclusions about IPS policy violations.

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 4-3 – 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] and section 7.2.2 of [IPSEP].

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 4-4 - 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 [IPSEP], the rationales provided in the prose in section 3 of [NDPP],in the tables in [NDPP] Annex A, section 5 of [FWEP] and section 7.3 of [IPSEP] 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 [IPSEP].

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 IPSEP. The definition of these components is given in [IPSEP]

- IPS_NTA_EXT.1
- IPS_IPB_EXT.1
- IPS_SBD_EXT.1
- IPS_ABD_EXT.1

5.1 Rationale for Extended Components

This ST includes these extended components to conform to the NDPP, FWEP and IPSEP 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], [FWEP] and [IPSEP].

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

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
Security Audit	FAU_GEN.1(1)	Audit Data Generation
	FAU_GEN.1(2)	Audit Data Generation (IPS)
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	External Audit Trail Storage
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4	Cryptographic Key Zeroization
	FCS_COP.1(1)	Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2), (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_PMG_EXT.1	Password Management
	FIA_UIA_EXT.1	User Identification and Authentication

	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_PSK_EXT.1	Extended: Pre-Shared Key Composition
Security Management	FMT_MTD.1	Management of TSF Data (for general TSF data)
	FMT_SMF.1(1), (2)	Specification of Management Functions
	FMT_SMR.2	Security Roles
Protection of the TSF	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_APW_EXT.1	Extended: Protection of Administrator Passwords
	FPT_STM.1	Reliable Time Stamps
	FPT_TUD_EXT.1	Extended: Trusted Update
	FPT_TST_EXT.1	TSF Testing
TOE Access	FTA_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	Inter-TSF trusted channel
	FTP_TRP.1	Trusted path
Stateful Traffic/Packet Filtering	FFW_RUL_EXT.1	Stateful Traffic Filtering
Intrusion Prevention System	IPS_NTA_EXT.1	Network Traffic Analysis
	IPS_IPB_EXT.1	IP Blocking
	IPS_SBD_EXT.1	Signature-Based IPS Functionality
	IPS_ABD_EXT.1	Anomaly-Based IPS Functionality

Table 6-1 – TOE Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1(1) Audit Data Generation

FAU_GEN.1.1(1) The TSF shall be able to generate an audit record of the following auditable events:

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

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

c) *All Administrative actions*;

d) [*Specifically defined auditable events listed in Table 6-2 - Audit Events and Details, and Table 6-3 - Audit Events and Details from FWEP*].

FAU_GEN.1.2(1) The TSF shall record within each audit record at least the following information:

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

b) For each audit event 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, and Table 6-3 - Audit Events and Details from FWEP*].

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FAU_GEN.1(1), (2)	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2), (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.
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Establishment/Termination of an IPsec SA.	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(1), (2)	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.	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 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

6.1.1.2 FAU_GEN.1(2) Audit Data Generation

FAU_GEN.1.1(2) **Refinement**: The TSF shall be able to generate an **IPS** audit record of the following auditable **IPS** events:

a) Start-up of the **IPS** functions;

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

~~c) All Administrative actions;~~

d) [All dissimilar IPS events;

e) All dissimilar IPS reactions;

f) Totals of similar events occurring within a specified time period; and

g) Totals of similar reactions occurring within a specified time period].

FAU_GEN.1.2(2) **Refinement**: The TSF shall record within each **IPS auditable event** record at least the following information:

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

b) For each **IPS** audit**able** event type, based on the auditable event definitions of the functional components included in the PP~~/ST~~, [*information specified in column three* Table 6-4 - Audit Events and Details from IPSEP.

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FMT_SMF.1(2)	Modification of an IPS policy element.	Identifier or name of the modified IPS policy element (e.g. which signature, baseline, or known-good/known-bad list was modified.
IPS_NTA_EXT.1	Modification of which IPS policies are active on a TOE interface. Enabling/disabling a TOE interface with IPS policies applied. Modification of which mode(s) is/are active on a TOE interface.	Identification of the TOE interface, and (when applicable) the IPS policy and interface mode.
IPS_IPB_EXT.1	Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.	Source and destination IP addresses (and, if applicable, indication of whether the source and/or destination address matched the list). TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset)

IPS_SBD_EXT.1	Inspected traffic matches a signature-based IPS policy.	Name or identifier of the matched signature. Source and destination IP addresses. The content of the header fields that were determined to match the signature. TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset)
IPS_ABD_EXT.1	Inspected traffic matches an anomaly-based IPS policy.	Source and destination IP addresses. The content of the header fields that were determined to match the policy. TOE interface that received the packet. Aspect of the anomaly-based IPS policy rule that triggered the event (e.g. throughput, time of day, frequency, etc.) Network-based action by the TOE (e.g. allowed, blocked, sent reset to source IP, sent blocking notification to firewall)

Table 6-4 - Audit Events and Details from IPSEP

6.1.1.3 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.

6.1.1.4 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 (FCS)

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

FCS_CKM.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 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.

6.1.2.2 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.

6.1.2.3 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* ***CBC*** and cryptographic key sizes 128-bits and 256-bits that meets the following:

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

6.1.2.4 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 or FIPS PUB 186-3, "Digital Signature Standard"**

6.1.2.5 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").**

6.1.2.6 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, SHA-512** and **message digest sizes 160, 256, 384, 512 bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

6.1.2.7 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, SHA-512**, **key size *160, 256, 512* (in bits) used in HMAC**, **and message digest sizes 160, 256, 512** bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

6.1.2.8 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 (256) seeded by an entropy source that accumulated entropy from a TSF-hardware-based noise source, and other independent TSF-hardware-based noise source.

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

6.1.2.9 Explicit: SSH (FCS_SSH_EXT)

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

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

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *32768 bytes* in an SSH transport connection are dropped.

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

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses SSH_RSA, ecdsa-sha2-nistp256 and ecdh-sha2-nistp384 as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512.

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

6.1.2.10 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

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-CBC-128 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC.

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.

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.

FCS_IPSEC_EXT.1.9 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.1.10 The TSF shall ensure that all IKE protocols perform peer authentication using a RSA, ECDSA algorithm and pre-shared keys.

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_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;

6.1.4.2 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.

6.1.4.3 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.

6.1.4.4 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.

6.1.4.5 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.

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, *conversion of 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 accept bit-based pre-shared keys.

6.1.5 Security Management (FMT)

6.1.5.1 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.

6.1.5.2 FMT_SMF.1(1) Specification of Management Functions (NDPP, FWEP)

FMT_SMF.1.1(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 the cryptographic functionality;*
- *Ability to configure Firewall rules (from FWEP);*
- *No other capabilities*.

6.1.5.3 FMT_SMF.1(2) Specification of Management Functions (IPSEP)

FMT_SMF.1.1(2) The TSF shall be capable of performing the following **security** management functions:

1) Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality
2) Modify these parameters that define the network traffic to be collected and analyzed:
    a) Source IP addresses (host address and network address)
    b) Destination IP addresses (host address and network address)
    c) Source port (TCP and UDP)
    d) Destination port (TCP and UDP)
    e) Protocol (IPv4 and IPv6)
    f) ICMP type and code
3) Update (import) signatures
4) Create custom signatures
5) Configure anomaly detection
6) Enable and disable actions to be taken when signature or anomaly matches are detected
7) Modify thresholds that trigger IPS reactions

8) Modify the duration of traffic blocking actions

9) Modify the known-good and known-bad lists (of IP addresses or address ranges)

10) Configure the known-good and known-bad lists to override signature-based IPS policies

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.1 Extended: 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.1 Extended: 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_STM.1 Reliable Time Stamps

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

6.1.6.4 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.

6.1.6.5 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.

6.1.7 TOE Access (FTA)

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.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 Inter-TSF Trusted Channel (Prevention of Disclosure)

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

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

6.1.8.2 FTP_TRP.1Trusted Path

FTP_TRP.1.1 **Refinement:** The TSF shall use **SSH, IPSec** 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 Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- 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.

6.1.10 Intrusion Prevention System (IPS)

6.1.10.1 IPS_NTA_EXT.1 Network Traffic Analysis

IPS_NTA_EXT.1.1 The TSF shall perform analysis of IP-based network traffic forwarded to the TOE's sensor interfaces, and detect violations of administratively-defined IPS policies.

IPS_NTA_EXT.1.2 The TSF shall process (be capable of inspecting) the following network traffic protocols:

- Internet Protocol (IPv4), RFC 791
- Internet Protocol version 6 (IPv6), RFC 2460
- Internet control message protocol version 4 (ICMPv4), RFC 792
- Internet control message protocol version 6 (ICMPv6), RFC 2463
- Transmission Control Protocol (TCP), RFC 793
- User Data Protocol (UDP), RFC 768

IPS_NTA_EXT.1.3 The TSF shall allow the signatures to be assigned to sensor interfaces configured for promiscuous mode, and to interfaces configured for inline mode, and support designation of one or more interfaces as 'management' for communication between the TOE and external entities without simultaneously being sensor interfaces.

- Promiscuous (listen-only) mode: *none*;
- Inline (data pass-through) mode: *Ethernet interfaces*;
- Management mode: [*assignment: console port, Ethernet interfaces, out-of-band management Ethernet interfaces*;
- Session-reset-capable interfaces: *Ethernet interfaces*;
- *and no other interface types*.

6.1.10.2 IPS_IPB_EXT.1 IP Blocking

IPS_IPB_EXT.1.1: The TSF shall support configuration and implementation of known-good and known-bad lists of source, destination IP addresses.

IPS_IPB_EXT.1.2: The TSF shall allow IPS Administrators and no other roles to configure IPS policy elements (known-good list rules, known-bad list rules, IP addresses).

6.1.10.3 IPS_SBD_EXT.1 Signature-Based IPS Functionality

IPS_SBD_EXT.1.1 The TSF shall support inspection of packet header contents and be able inspect at least the following header fields:

- IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options.
- IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.
- ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code).
- ICMPv6: Type; Code; and Header Checksum.
- TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: Source port; destination port; length; and UDP checksum.

IPS_SBD_EXT.1.2 The TSF shall support inspection of packet payload data and be able to inspect at least the following data elements to perform string-based pattern-matching:

- ICMPv4 data: characters beyond the first 4 bytes of the ICMP header.
- ICMPv6 data: characters beyond the first 4 bytes of the ICMP header.
- TCP data (characters beyond the 20 byte TCP header), with support for detection of:
  i. FTP (file transfer) commands: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type.

ii. HTTP (web) commands and content: commands including GET and POST, and administrator-defined strings to match URLs/URIs, and web page content.

iii. SMTP (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.

- UDP data: characters beyond the first 8 bytes of the UDP header;

In addition, the TSF shall support stream reassembly or equivalent to detect malicious payload even if it is split across multiple non-fragmented packets.

IPS_SBD_EXT.1.3: The TSF shall be able to detect the following header-based signatures (using fields identified in IPS_SBD_EXT.1.1) at IPS sensor interfaces:

a) IP Attacks

i) IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)

ii) IP source address equal to the IP destination (Land attack)

b) ICMP Attacks

i) Fragmented ICMP Traffic (e.g. Nuke attack)

ii) Large ICMP Traffic (Ping of Death attack)

c) TCP Attacks

i) TCP NULL flags

ii) TCP SYN+FIN flags

iii) TCP FIN only flags

iv) TCP SYN+RST flags

d) UDP Attacks

i) UDP Bomb Attack

ii) UDP Chargen DoS Attack

IPS_SBD_EXT.1.4: The TSF shall be able to detect all the following traffic-pattern detection signatures, and to have these signatures applied to IPS sensor interfaces:

a) Flooding a host (DoS attack)

i) ICMP flooding (Smurf attack, and ping flood)

ii) TCP flooding (e.g. SYN flood)

b) Flooding a network (DoS attack)

c) Protocol and port scanning (Reconnaissance attacks that scan target IP addresses for open/listening/responsive services by targeting multiple protocols/ports on one or more target IP address using obvious (sequentially numbered) patterns of target protocol/port numbers of by randomizing the protocol/port numbers and/or randomizing the time delays between transmissions)

i) IP protocol scanning

ii) TCP port scanning

iii) UDP port scanning

iv) ICMP scanning

IPS_SBD_EXT.1.5 The TSF shall allow the following operations to be associated with signature-based IPS policies:

- In any mode, for any sensor interface:
  - allow the traffic flow
- In inline mode:
  - allow the traffic flow
  - block/drop the traffic flow
  - and no other actions

6.1.10.4 IPS_ABD_EXT.1 Anomaly-Based IPS Functionality

IPS_ABD_EXT.1.1 The TSF shall support the definition of anomaly ('unexpected') traffic patterns including the specification of:

- throughput (*bits per second*);
- time of day;
- and no other methods;

and the following network protocol fields:

- *IPv4: source Address; destination address.*

- *IPv6: source address; destination address.*
- *TCP: source port; destination port.*
- *UDP: source port; destination port.*

IPS_ABD_EXT.1.2 The TSF shall support the definition of anomaly activity through: manual configuration by administrators.

IPS_ABD_EXT.1.3 The TSF shall allow the following operations to be associated with anomaly-based IPS policies:

- In any mode, for any sensor interface:
  - allow the traffic flow
- In inline mode:
  - allow the traffic flow
  - block/drop the traffic flow
  - and no other actions

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 IPSEP. Any hierarchies and dependencies are satisfied in accordance with the NDPP, FWEP and IPSEP.

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 IPSEP 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 of IPSEP.

6.4.2 Sufficiency of Security Requirements

The following tables present a mapping of the rationale of TOE Security Requirements to Objectives as described in the NDPP Section 3, FWEP Section 4 and IPSEP Section 3

OBJECTIVE	SFR
Protected Communications O.PROTECTED_COMMUNICATIONS	FCS_CKM.1 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 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(1) FAU_GEN.2 FAU_STG_EXT.1 FPT_STM.1
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(1) FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4 FMT_SMR.2 FPT_SKP_EXT.1 FPT_APW_EXT.1 FTA_TAB.1
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(1) FFW_RUL_EXT.1
O.TOE_ADMINISTRATION	FMT_SMF.1(1)

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

OBJECTIVE	SFR
O.SYSTEM_MONITORING	FAU_GEN.1((2) IPS_NTA_EXT.1 IPS_IPB_EXT.1 IPS_SBD_EXT.1 IPS_ABD_EXT.1
O.IPSANALYZE	IPS_NTA_EXT.1 IPS_IPB_EXT.1 IPS_SBD_EXT.1 IPS_ABD_EXT.1
O.IPSREACT	IPS_ABD_EXT.1.3 IPS_SBD_EXT.1.5
O.TOE_ADMINISTRATION	FMT_SMF.1(2)

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

6.4.3 Security Assurance Requirements

The assurance security requirements for this Security Target are from the NDPP, FWEP, and IPSEP. 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
	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 IPSEP Section 5.

6.4.4 Security Assurance Requirements Rationale

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

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	Security Target: Junos 12.3 X48-D30 for SRX XLR Platforms (NDPP, TFFWEP, IPSEP)
AGD_OPE.1 Operational User Guidance AGD_PRE.1 Preparative Procedures	Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices Release 12.3X48-D30
ALC_CMC.1 Labeling of the TOE ALC_CMS.1 TOE CM Coverage	Security Target: Junos 12.3 X48-D30 for SRX XLR Platforms (NDPP, TFFWEP, IPSEP)
ATE_IND.1 Independent Testing	Provided by Evaluation Lab
AVA_VAN.1 Vulnerability Analysis	Provided by Evaluation Lab

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
- Intrusion Prevention System

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, Table 6-3, **Error! Reference source not found.** and Table 6-4):

- Start-up and shutdown of the audit function;
- Configuration is committed;
- Configuration is changed;
- All use of the identification and authentication mechanisms;
- Service requests;
- Failure to establish an SSH session and establishment/termination of an SSH session;
- Changes to the time;

- Initiation of update;
- Indication that TSF self-test was completed;
- Any attempts at unlocking of an interactive session;
- Termination of a remote session by the session locking mechanism;
- Termination of an interactive session;
- Initiation/termination/failure of the trusted path/channel functions.
- Application of firewall rules configured with the ‘log’ operation by the stateful traffic filtering function;
- Indication of packets dropped due to too much network traffic by the stateful traffic filtering function;
- Session establishment with peer;
- Establishing session with CA;
- Application of rules configured with the ‘log’ operation by the packet filtering function;
- Indication of packets dropped due to too much network traffic by the packet filtering function;
- Start-up of the IPS functions;
- All dissimilar IPS events and reactions;
- Totals of similar events and reactions occurring within a specified time period;
- Modification of an IPS policy element
- Modification of which IPS policies are active on a TOE interface
- Enabling/disabling a TOE interface with IPS policies applied
- Modification of which mode(s) is/are active on a TOE interface
- Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy
- Inspected traffic matches a signature-based IPS policy
- Inspected traffic matches an anomaly-based IPS policy

The TOE records the following with each log entry:

- date and time of the event and/or reaction,
- type of event and/or reaction,
- subject identity (where applicable),
- the outcome (success or failure) of the event (where applicable);

Additional information is recorded for certain audit events as per Table 6-2, Table 6-3 and Table 6-4.

The TOE generates event logs when a firewall or IPS – also referred to as Intrusion Detection and Prevention (IDP) in the Junos OS literature – rules are triggered. Event logging can be configured on a rule-by-rule basis when defining individual firewall and IPS policies. Because of the nature of IDP event logs, log generation often happens in bursts and can generate a much larger volume of messages during an attack. To manage the volume of log messages, Junos supports log suppression, which suppresses multiple instances of the same log occurring from the same or similar sessions over the same period of time. IDP log suppression is enabled by default and can be customized based on configurable attributes:

- Source/destination addresses;
- Number of log occurrences after which log suppression begins;
- Maximum number of logs that log suppression can operate on;
- Time after which suppressed logs are reported.

Suppressed logs are reported as single log entry containing the count of occurrences.

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). By default, the TOE stores all logs locally. The level of what to log locally is configurable. Local audit log are stored in /var/log/. When the TOE is configured to direct syslog traffic to an external syslog server via a IPSec, a VPN tunnel is initiated from the TOE immediately upon configuration commit and communications from TOE to the syslog server is encrypted and integrity protected. Audit records are sent to the syslog server periodically as configured by the Administrator

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'. 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. 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.

For more information about configuring event logging, see the *Junos OS Complete Software Guide for SRX Series Services Gateways (Volume 1), Chapter 35 'System Log Monitoring and Troubleshooting Guide for Security Devices'* and the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices*.

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

- FAU_GEN.1(1), (2)
- 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 by allowing the administrator to enable the appropriate cryptographic functions.

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 12.3X48. – Data Plane	4070, 4069, 4068, 4067, 4066
AES-CBC	Junos FIPS Version 12.3X48 – Data Plane	4070, 4069, 4068, 4067, 4066
AES-CBC	Junos FIPS Version 12.3X48 – Authentec	4055

AES-CBC	Junos FIPS Version 12.3X48 – OpenSSL	4056
SHA	Junos FIPS Version 12.3X48 – Data Plane	3353, 3352, 3351, 3350, 3349
SHA	Junos FIPS Version 12.3X48 – Authentec	3342
SHA	Junos FIPS Version 12.3X48 – OpenSSL	3343
HMAC-SHA	Junos FIPS Version 12.3X48 – Data Plane	2657, 2656, 2655, 2654, 2653
HMAC-SHA	Junos FIPS Version 12.3X48 – Authentec	2647
HMAC-SHA	Junos FIPS Version 12.3X48 – OpenSSL	2648
HMAC_DRBG	Junos FIPS Version 12.3X48 – OpenSSL	1216
ECDSA	Junos FIPS Version 12.3X48 –Authentec	917, 916, 915, 914, 913, 912
ECDSA	Junos FIPS Version 12.3X48 – Authentec KeyGen	917, 916, 915, 914, 913, 912
ECDSA	Junos FIPS Version 12.3X48 – OpenSSL	909
ECDSA	Junos FIPS Version 12.3X48 – OpenSSL KeyGen	909
DSA	Junos FIPS Version 12.3X48 – Authentec KeyGen	1104, 1103, 1102, 1101, 1100, 1099
DSA	Junos FIPS Version 12.3X48 – OpenSSL KeyGen	1096

RSA	Junos FIPS Version 12.3X48 – Authentec	2202, 2201, 2200, 2199, 2198, 2197
RSA	Junos FIPS Version 12.3X48 – OpenSSL KeyGen	2087

Table 7-1 – CAVS Certificate Results

All random number generation by the TOE is performed in accordance with NIST Special Publication 800-90 using HMAC_DRBG. (FCS_RBG_EXT.1.1)

Asymmetric keys are generated in accordance with NIST SP 800-56A and NIST SP800-56B for use with SSH to the admin console. Asymmetric keys are generated in accordance with NIST SP 800-56A and FIPS PUB 186-3 for IKE with IPSec . The TOE complies with section 5.6 of NIST SP 800-56A regarding asymmetric key pair generation. The TOE complies with section 6 of NIST SP 800-56B regarding RSA key pair generation. The 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. (FCS_CKM.1)

The TOE handles zeroization for all CSP, plaintext secret and private cryptographic keys according to the table below. The zeroization mechanisms of sensitive data stored in RAM entails overwriting the data with zeros. Sensitive data on non-volatile storage is zeroized when the explicit “request system zeroize” command is executed, which overwrites the data three times, first with the byte pattern 0xff, then 0x00, and then 0xff again. (FCS_CKM_EXT.4.1)

CSP	Description	How Stored	Where Stored	Zeroization Method
SSH Private Host Key	The first time SSH is configured, the key is generated. RSA 2048, ECDSA P-256, ECDSA P-384. Used to identify the host.	Plaintext	Disk/Memory	Keys in RAM are zeroized at reboot time. Keys in disk are zeroized when the command "request system zeroize" is executed.
SSH Session Key	Session keys used with SSH. AES, HMAC, HMAC-SHA-2	Plaintext	Memory	Keys in RAM are zeroized at reboot time.
SSH DH	DH ephemeral private exponents used in SSH. DH (N = 256 bit, 320 bit, 384 bit, 512 bit, or 1024 bit), ECDH P-256, or ECDH P-384	Plaintext	Memory	Keys in RAM are zeroized at reboot time.
User Password	Plaintext value as entered by user for authentication	Hashed	Disk/Memory	Zeroized when the command "request system zeroize" is executed.
RNG State	Internal state and seed key of HMAC_DRBG	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.

IKE Private Host key	Private authentication key used in IKE. RSA 2048, ECDSA P-256, ECDSA P-384	Plaintext	Disk/Memory	'clear security IKE security-association' command or reboot the box. Private keys stored in flash are not zeroized unless an explicit "request system zeroize" is executed.
IKE-SKEYID	IKE master secret used to derive IKE and IPsec ESP session keys	Plaintext	Memory	'clear security IKE security-association' command or reboot the box
IKE Session Keys	IKE session key. AES, HMAC	Plaintext	Memory	'clear security IKE security-association' command or reboot the box
ESP Session Key	ESP session keys. AES, HMAC	Plaintext	Memory	'clear security ipsec security-assocation' or reboot the box.
IKE-DH Private Exponent	Ephemeral DH private exponent used in IKE. DH N = 224 bit, ECDH P-256, or ECDH P-384	Plaintext	Memory	'clear security IKE security-association' command or reboot the box.

IKE-PSK	Pre-shared authentication key used in IKE.	Hashed	Disk/Memory	'clear security IKE security-association' command or reboot the box. Key values stored in flash are not zeroized unless an explicit "request system zeroize" is executed.

Table 7-2 - Key Zeroization Handling

7.3.1 SSH Support

The TOE uses SSH_RSA and SSH-ECDSA as its public key algorithms for authentication, as well as password-based authentication for SSH The TOE implements a timeout period for authentication for the SSHv2 protocol and provides a limit of three failed authentication attempts. (FCS_SSH_EXT.1.1, FCS_SSH_EXT.1.2, FCS_SSH_EXT.1.5)

The TOE supports AES-CBC-128 and AES-CBC-256 encryption algorithms for SSH transport. (FCS_SSH_EXT.1.4)

The data integrity algorithms used in SSH transport connection are HMAC-SHA1, HMAC-SHA1-96, as required by RFC4253, and HMAC-SHA2-256, AND HMAC-SHA2-512 as recommended by RFC6668 (FCS_SSH_EXT.1.6).

The OpenSSH implementation of HMAC-SHA1-96 uses HMAC-SHA1 as its underlying component, which has been implemented and tested per CAVP certificate #2648

Key exchange is performed using diffie-hellman-group14-sha1 as per RFC4253 and ecdh-sha2-nistp256, ecdh-sha2-nistp384 as per RFC5656 (FCS_SSH_EXT.1.7).

Packets greater than 32768 bytes in an SSH transport connection are dropped and the connection is terminated by the TOE. The SSH daemon maintains a 32768 byte 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. (FCS_SSH_EXT.1.3)

7.3.2 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-CBC-128 or AES-CBC-256 using HMAC SHA-1 and SHA-256. Keyed-hash algorithms including HMAC-SHA1-96, HMAC-SHA-256-128 can be configured for AES-CBC. (FCS_IPSEC_EXT.1.4)

IKEv1 and IKEv2 are implemented. IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109 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 AES-CBC-128, AES-CBC-256for payload protection in IKEv1 and IKEv2. (FCS_IPSEC_EXT.1.6)

In the evaluated configuration, the TOE permits only main mode to be configured for IKEv1 Phase 1 exchanges. There is no option to configure aggressive mode. (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 supports Diffie-Hellman Groups 14, 19, 20, and 24. In the IKEv1 phase 1 and phase 2 exchanges, the TOE and peer will agree on the best DH group both can support.When the TOE receives an IKE proposal, it will select the first DH group that matches the acceptable DH groups configured in the TOE (one or more of DH Groups 14, 19, 20 or 24) and the negotiation will fail if there is no match. Similarly, when the peer initiates the IKE protocol, the TOE will select the first match from the IKE proposal sent by the peer and the negotiation fails is no acceptable match is found. (FCS_IPSEC_EXT.1.9)

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

The TOE validates X.509v3 certificates from the peer by confirming that the ID payload passed in IKE matches the identifiers in the peer certificate. The TOE also verifies that the signature is correct, based on the root CA public key.

The TOE also validates the certificate based on its time window, so correct UTC time on the router is essential. In addition to the certificate checks, the TOE confirms that message data received from the peer has the correct signature based on the peer's public key as found in its certificate. (FCS_IPSEC_EXT.1.10)

For more information about SSH and IPSec support, see the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices; Junos OS Complete Software Guide for SRX Series Services Gateways (Volume 2), Part 19 'VPN Feature Guide for Security Devices'* and *Junos OS Complete Software Guide for SRX Series Services Gateways (Volume 1), Part 5 'Administration Guide for Security Devices*'.

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

- FCS_CKM.1
- FCS_CKM_EXT.4
- FCS_COP.1(1)
- FCS_COP.1(2),(3),(4),(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". 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. (FIA_PMG_EXT.1)

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

[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

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, 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 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. (FIA_UAU.7)

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. (FIA_UIA_EXT.1)

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

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

configured maximum, no authentication attempts are accepted for a configured time interval. When the interval has expires, authentication attempts are again accepted.

Certificates are stored in non-volatile flash memory. Access to flash memory requires administrator credentials. A certificate may be loaded via command line.

The TOE uses pre-shared keys for IPSec. 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 bit-based 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, the administrator uses the CLI command

```
request security pki generate-certificate-request
```

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

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_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.2
- FIA_UAU.7
- 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. Additionally, an IPSec tunnel can be established between the remote administrator and the TOE, which further protects the confidentiality and integrity of the SSH communication, as well as authenticates both end-points, using either a pre-shared key or RSA/ECDSA certificates (see Section 0). 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 IPSec-associated cryptographic functionality;
- Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality
- Modify these parameters that define the network traffic to be collected and analyzed:

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

  - Source IP addresses (host address and network address);
  - Destination IP addresses (host address and network address);
  - Source port (TCP and UDP);
  - Destination port (TCP and UDP);
  - Protocol (IPv4 and IPv6)
  - ICMP type and code

- Update (import) IPS signatures;
- Create custom IPS signatures;
- Configure anomaly detection;
- Enable and disable actions to be taken when signature or anomaly matches are detected;
- Modify thresholds that trigger IPS reactions;
- Modify the duration of traffic blocking actions;
- Modify the known-good and known-bad lists (of IP addresses or address ranges);
- Configure the known-good and known-bad lists to override signature-based IPS policies. (FMT_SMF.1)

Detailed topics on the secure management of the TOE are discussed in *Junos OS Complete Software Guide for SRX Series Services Gateways (Volumes 1 and 2)* and the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices*.

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

- FMT_MTD.1
- FMT_SMF.1(1), (2)
- 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:

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 11 of the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices*.

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.

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

- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_STM.1
- FPT_TUD_EXT.1
- FPT_TST_EXT.1

7.8 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 IPSec.

It also supports Trusted Paths between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification using SSHv2, which can be further protected using IPSec.

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

- FTP_ITC.1
- FTP_TRP.1

7.10 Stateful Traffic/Packet Filtering FWEP

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 (flowd) 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 IPS 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

For more information about configuring event logging, see the *Junos OS Complete Software Guide for SRX Series Services Gateways (Volume 2), Part 4: ‘Building Blocks Feature Guide for Security Devices’* and the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices*.

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

- FFW_RUL_EXT.1

7.11 Intrusion Prevention System

The Junos OS Intrusion Detection and Prevention (IDP) policy enables selectively enforcing various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. Policy rules can be defined to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy is made up of rule bases, and each rule base contains a set of rules that specify rule parameters, such as traffic match conditions, action, and logging requirements. IDP policies can then be associated to firewall policies. IDP can be invoked on a firewall rule by rule basis for maximum granularity. Only firewall policies marked for IDP will be processed by IDP engine, all other rules will only be processed by the firewall[4].

Firewall Policies match Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface and VLAN matching can be achieved through the use of zones. Rules are organized into a firewall policy rulebase. Within IPS Policies, further matching for specific attacks is done on Source Zone, Destination Zone, Source IP, Destination IP, Source Port, Destination Port, and Protocol. Interface matching can be achieved through the use of zones. Attack Actions are configurable on a rule by rule basis. Rules within policies are processed in an Administrator-defined order when network traffic flows through the TOE network interfaces. (IPS_NTA_EXT.1.1)

Once stateful firewall processing of packets has been performed by the Information Flow subsystem, if a firewall policy that has been marked for IDP processing is triggered, the packets are processed by the IPS subsystem as follows:

- Fragmentation Processing – IP Fragments are reordered and reassembled. Duplicate, over/undersized, overlapping, incomplete and other invalid fragments are discarded.
- Flow Module SSL Decryption – sessions are checked for existing IP Actions, if none exist, new sessions are created. If a destination is marked for SSL decryption, a copy of the SSL traffic will be sent to the decryption engine. The original packet will be queue until inspection is complete.
- Packet Serialization and TCP Reassembly – packets are ordered and all TCP packets are reassembled into complete application messages.

[4] Note that some of the security functionality required by the IPS EP is implemented at the firewall level without intervention of Junos IDP engine.

- Application ID – pattern matching is performed on the traffic to determine what application the traffic is. The traffic is still inspected for Attacks, even if application cannot be determined.
- Protocol Decoding – protocol parsing and decoding is performed. Messages are deconstructed into application "contexts" which identify components of messages. Protocol Anomaly Detection is performed, along with AppDoS (if configured) by thresholds of these contexts.
- Attack Signature Matching – signatures are detected via deterministic finite automaton (DFA) pattern matching.
- IDP Attack Actions – when an attack is detected the corresponding policy configured action is executed. Possible actions include:
    - No Action
    - Drop packet
    - Drop connection
    - Close client (send an RST packet to the client)
    - Close server (sends an RST packet to the server)
    - Close client and server (sends an RST packet to both client and server)

The TOE supports stateful signature based attack detection defined as Attack Objects. Attack Objects use context based matching to match regular expressions in specific locations where they occur. Attack Objects can be composed of multiple signatures and protocol anomalies, including logical expressions between signatures for compound matching.

As indicated in Section 7.10 the TOE is capable of inspecting IPv4, IPv6, ICMPv4, TCP and UPD traffic. Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team. (IPS_NTA_EXT.1.2)

The TOE is capable of inspecting all traffic passing through the TOE's Ethernet interfaces (inline mode). Ethernet interfaces can be assigned to Zones on which firewall and IDP policies are predicated. The TOE supports management through the console port, as well as through a dedicated Ethernet management port whose traffic is never processed for routing. Remote management of the TOE can also be performed via SSH as described in 7.6. (IPS_NTA_EXT.1.3)

The TOE supports the definition of known-good and known-bad lists of source and/or destination addresses at the firewall rule level as described in Section 7.10. Address ranges can be defined by creating address book entries and attaching them to firewall policies. (IPS_IPB_EXT.1)

IPS signatures (in the sense of the IPS EP) are articulated at different points along the traffic processing flow implemented in the TOE. In Junos OS, interfaces are grouped into zones. The TOE supports the definition of signatures at the zone level, also known as the screen level. Junos OS screen options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. Sanity checks on IPv4 and IPv6 aimed at detecting malformed packets are performed at the screen level. In addition to attack detection and prevention at the screen level, Junos OS implements firewall and IDP policies at the inter-, intra-, and super-zone policy levels (super-zone here means in global policies, where no security zones are referenced). The TOE supports inspection of the following packet header information:

- IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options.
- IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.
- ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code).
- ICMPv6: Type; Code; and Header Checksum.
- TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: Source port; destination port; length; and UDP checksum.

Signatures can be defined to match the any of above header-field values, using the command "set security idp custom-attack", along with the actions (allow/block), using the command "set security idp idp-policy", that the TOE will perform when a

match is found in the processed packets. The matching criteria can be "equal", "greater-than", "less-than" or "not-equal". (IPS_SBD_EXT.1.1)

The TOE also supports string-based pattern-matching inspection of packet payload data for the above listed protocols. For TCP payload inspection, Junos OS provides pre-defined attack signatures to detect FTP commands, HTTP commands and content, and STMP states. Alternative, administrators can define custom-attack signatures for these application layer protocols using the command “set security idp custom-attack”. (IPS_SBD_EXT.1.2)

The TOE is capable of detecting the following signatures using Junos predefined screen options:

IPS EP signature name	Junos screen name
IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)	ip tear-drop
IP source address equal to the IP destination (Land attack)	tcp land
Fragmented ICMP Traffic (e.g. Nuke attack)	icmp fragment
Large ICMP Traffic (Ping of Death attack)	icmp ping-death
TCP NULL flags	tcp tcp-no-flag
TCP SYN+FIN flags	tcp syn-fin
TCP FIN only flags	tcp fin-no-ack
UDP Bomb Attack	udp length-error
ICMP flooding (Smurf attack, and ping flood)	icmp flood
TCP flooding (e.g. SYN flood)	tcp syn-flood
IP protocol scanning	ip unknown-protocol
TCP port scanning	tcp port-scan
UDP port scanning	udp port-scan
ICMP scanning	icmp ip-sweep

The default action for the above screens is to drop the packets. To allow the packets through, the “alarm-without-drop” action can be defined using the command “set security screen ids-option”.

The TOE is also capable of detecting the following signatures:

- TCP SYN+RST flags, by defining an custom attack to match “protocol tcp tcp-flags rst” and “protocol tcp tcp-flags syn”[5];
- UDP Chargen DoS attack , by configuring a firewall policy to match the predefined “junos-chargen” with the desired allow/block reaction;
- Flooding of a network (DoS attack), by the configuration of policers that allow establishing prioritization and bandwidth limits for different type of network traffic. (IPS_SBD_EXT.1.3, IPS_SBD_EXT.1.4)

The TOE allows administrators to define signatures for anomalous traffic in terms of throughput (bits per second) and time of the day for defined source/destination address and source/destination port.

Anomaly signatures based on time of day characteristics are implemented by configuring schedulers using the Junos command ‘set schedulers’ and attaching them to firewall policies, which in turn specify the target traffic in terms of IP addresses and port numbers as well as the action to be perform on signature triggering (allow or block/drop traffic).

Anomaly signatures based on throughput characteristics are implemented by configuring policers with a bandwidth limit and the desired signature action (discard or forward), using the Junos command ‘set firewall policer’, and attaching it to any interface with the Junos command ‘set interfaces’. Traffic exceeding the specified throughput limit is dropped when the policer is configured to discard traffic. A policer can be applied to specific inbound or outbound IP packets in a Layer 3 traffic flow at a logical interface by using a stateless firewall filter. If an input firewall filter is configured on the same logical interface as a policer, the policer is executed first. If an output firewall filter is configured on the same logical interface as a policer, the firewall filter is executed first. (IPS_ABD_EXT.1)

For more information about configuring event logging, see the *Junos OS Complete Software Guide for SRX Series Services Gateways (Volume 2), Part 9: ‘Intrusion Detection and Prevention Feature Guide for Security Devices’* and the *Junos OS Common Criteria Evaluated Configuration Guide for SRX Series Security Devices*.

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

[5] By default the TOE will drop packets where the TCP flags SYN and ACK are set at the screen level.

- IPS_NTA_EXT.1
- IPS_IPB_EXT.1
- IPS_SBD_EXT.1
- IPS_ABD_EXT.1