Security Target Junos OS 20.3R3 for NFX350

Document Version: 1.2

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History:

Version	Date	Changes
Version 0.1	26 May 2021	Initial draft
Version 0.2	28 June 2021	Addressed review comments
Version 0.3	16 December 2021	Updated TDs
Version 1.0	06 May 2022	Updated TDs and addressed review comments
Version 1.1	13 June 2022	Addressed review comments
Version 1.2	20 June 2022	Minor updates based on ECR comments

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	Security Target Junos 20.3R3 for NFX350
ST Version	1.2
ST Date	20 June 2022
ST Author	Acumen Security, LLC.
TOE Identifier	Junos OS 20.3R3 for NFX350
TOE Version	20.3R3
TOE Developer	Juniper Networks, Inc.
Key Words	Network Device, VPN Gateways, IPS, Firewall

1.2 TOE Overview

The TOE is Juniper Networks, Inc. Junos OS 20.3R3 for NFX350 Network Services Platform. The NFX350 is a network device that integrates routing, switching, and security functions on a single platform.

The NFX350 supports the definition of, and enforces, information flow policies among network nodes, also providing for stateful inspection of every packet that traverses the network and 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 provides multi-site virtual private network (VPN) gateway functionality, and also implements Intrusion Prevention System functionality, capable of monitoring information flows to detect potential attacks based on pre-defined attack signature and anomaly characteristics in the traffic.

The deployment of the Junos OS 20.3R3 for NFX350 TOE includes a hypervisor, which runs a virtual machine (VM) on an NFX350 series hardware model:

- NFX350-S1
- NFX350-S2
- NFX350-S3

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

The TOE includes a Linux Operating System (OS), Junos Control Plane (JCP), a Juniper Device Manager (JDM) and an Open vSwitch (OVS) bridge. NFX350 supports the installation of 3rd party VMs and containers, but installation of 3rd party VMs and containers is not allowed in the evaluated configuration. Figure 1 below shows the general architecture for the NFX350.

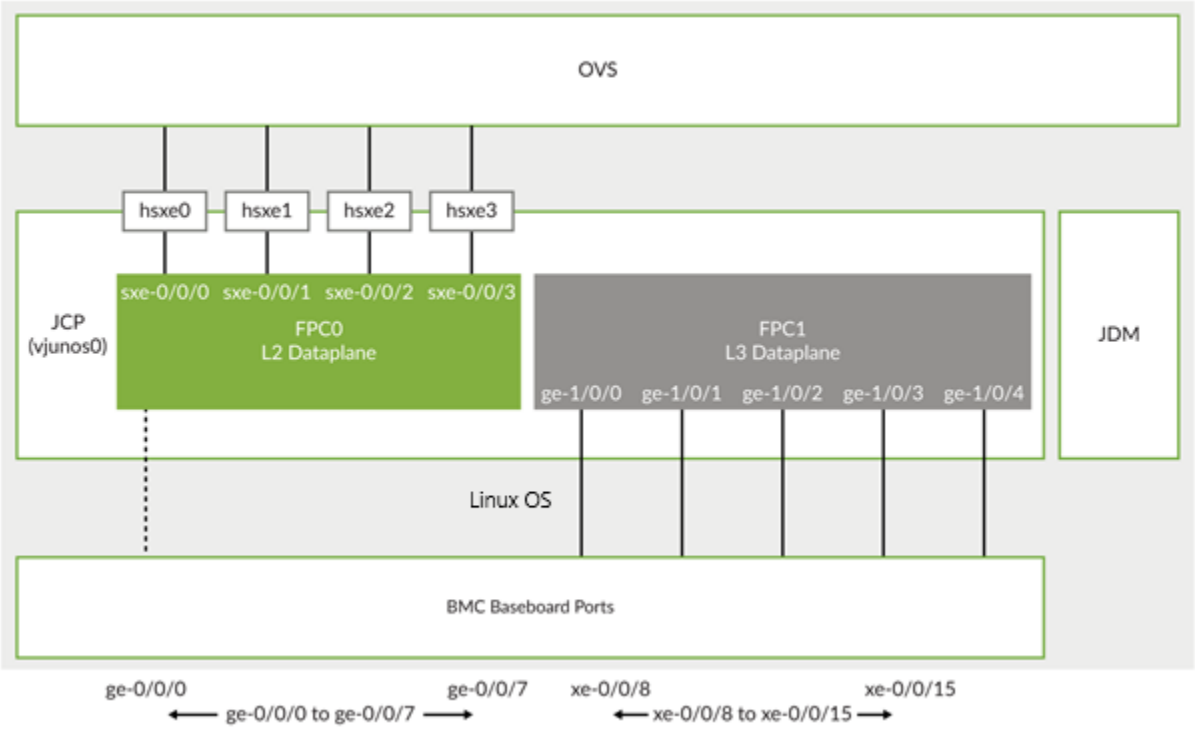

Figure 1 NFX350 Architecture

1.3.1 Linux OS

NFX350 is running on Wind River Linux 8 as its host OS. The host OS functions as a hypervisor and runs natively on an Intel Xeon D processor.

1.3.2 Junos Control Plane

Junos Control Plane (JCP) is the Junos VM running on the host OS. JCP is used to configure the network ports of the NFX350 device, and JCP runs by default as vjunos0 on NFX350. The JCP functions as the single point of management for all the components. The JCP supports:

- Layer 2 to Layer 3 routing services
- Layer 3 to Layer 4 security services
- Layer 4 to Layer 7 advanced security services

In addition, the JCP enables virtualized network functions (VNF) lifecycle management. VNF is a virtualized implementation of a network device and its functions. In the NFX350 NextGen architecture, Linux functions as the hypervisor, and it creates and runs the VNFs. The VNFs include functions such as firewalls, routers, and WAN accelerators.

The JCP VM is the single administration point for the NFX350 platform. It is the front-end for all functionality provided by the NFX350 software. Logging in via console of SSH take the user to a CLI prompt on the JCP VM. This CLI is the single point of configuration for all NFX350 services.

1.3.1.1 L2 Data Plane

L2 data plane manages the Layer 2 traffic. The L2 data plane forwards the LAN traffic to the OVS bridge. The L2 data plane is mapped to the virtual FPC0 on the JCP.

1.3.1.2 L3 Data Plane

L3 data plane provides data path functions for the Layer 3 to Layer 7 services. The L3 data plane is mapped to the virtual FPC1 on the JCP.

1.3.3 Juniper Device Manager (JDM)

JDM is an application container that manages VNFs and provides infrastructure services. The JDM functions in the background. JDM is a low-footprint Linux container that provides these functions:

- Virtual Machine (VM) lifecycle management
- Device management and isolation of host OS from user installations
- NIC, single-root I/O virtualization (SR-IOV), and virtual input/output (VirtIO) interface provisioning
- Inventory and resource management
- Internal network and image management
- Service chaining—provides building blocks such as virtual interfaces and bridges for users to implement service chaining polices
- Virtual console access to VNFs including vSRX and vjunos

1.3.4 Open vSwitch (OVS) bridge

The OVS bridge is a VLAN-aware system bridge that acts as the network functions virtualization backplane to which the VNFs, FPC1, and FPC0 connect.

1.3.5 NFX350 Hardware

The hardware model specifications are described in the table below.

Table 2 – TOE Hardware Specifications

Specification	NFX350-S1	NFX350-S2	NFX350-S3
Dimensions (H x W x D)	1.72 x 17.32 x 20.86 in (4.37 x 44.0 x 53.0 cm)	1.72 x 17.32 x 20.86 in (4.37 x 44.0 x 53.0 cm))	1.72 x 17.32 x 20.86 in (4.37 x 44.0 x 53.0 cm))
Rack units (U)	1 U	1 U	1 U
Weight	18.5 lb (8.4 kg)	18.6 lb (8.45 kg)	18.6 lb (8.45 kg)
Airflow	Front-to-back (AFO) forced cooling	Front-to-back (AFO) forced cooling	Front-to-back (AFO) forced cooling
Acoustics	61 dBA	61 dBA	61 dBA

Specification	NFX350-S1	NFX350-S2	NFX350-S3
Power	650W hot-swappable AC-DC/DC-DC	650W hot-swappable AC-DC/DC-DC	650W hot-swappable AC-DC/DC-DC
CPU	Intel Xeon D-2146NT 8 Core	Intel Xeon D-2166NT 12 Core	Intel Xeon D-2187NT 16 Core
Micro-Architecture	Skylake	Skylake	Skylake
Memory	32 GB DDR4	64 GB DDR4	128 GB DDR4
Storage	100 GB SSD[1]	100 GB SSD[1]	100 GB SSD[1]
Software	Wind River Linux 8	Wind River Linux 8	Wind River Linux 8
Network interfaces	• 8 x 10/100/ 1000BASE-T RJ-45 LAN or WAN ports • 8 x 1GbE/10GbE SFP+ LAN or WAN ports • 1 x 10/100/ 1000BASE-T RJ-45 management port	• 8 x 10/100/ 1000BASE-T RJ-45 LAN or WAN ports • 8 x 1Gb[2]E/10GbE SFP+ LAN or WAN ports • 1 x 10/100/ 1000BASE-T RJ-45 management port	• 8 x 10/100/ • 1000BASE-T RJ-45 LAN or WAN ports • 8 x 1GbE/10GbE • SFP+ LAN or WAN ports • 1 x 10/100/ • 1000BASE-T RJ-45 management port
Managed Secure Router[2]	12 Gbps	20 Gbps	30 Gbps
Managed Security[2]	12 Gbps	20 Gbps	30 Gbps
IPsec[2]	2.5 Gbps	5 Gbps	7.5 Gbps
Out-of-band interfaces	RJ-45 console port Mini USB console port USB 2.0 port	RJ-45 console port Mini USB console port USB 2.0 port	RJ-45 console port Mini USB console port USB 2.0 port
Maximum number of VNFs	8	10	12

The JCP's FreeBSD kernel uses the kernel-based virtual machine (KVM) as a virtualization infrastructure. KVM is part of the standard NFX350 distribution and can be used to create multiple VMs and to install security and networking appliances. The TOE uses Open vSwitch as a backplane between these VMs. However, in the TOE evaluated configuration, only a single VM is running and no security or networking appliances may be installed. Therefore, in the evaluated configuration the KVM functions simply as a pass-through layer.

The interfaces on the NFX350 devices include physical interfaces and virtual interfaces.

The physical interfaces represent the physical ports on the NFX350 chassis. The physical interfaces include network and management ports:

[1] Raw capacity; actual capacity will be lower due to overprovisioning.
[2] Maximum throughput mode.

- Network ports NFX350 chassis —
    - 8 x 10/100/1000BASE-T RJ-45 LAN or WAN ports
    - 8 x 1GbE/10GbE SFP+ LAN or WAN ports
    - 1 x 10/100/1000BASE-T RJ-45 management port
- Management port – NFX350 device has a dedicated management ports which functions as the out-of-band management interface –
    - RJ-45 console port
    - Mini USB console port
    - 2 x USB 3.0 port

Each physical NIC port has sixteen virtual functions (VFs) enabled by default.

The virtual interfaces on the NFX350 device include the following:

- Virtual layer 3 interfaces – used to configure layer 3 features such as routing protocols and QoS
- Virtual SXE interfaces – four SXE ports (static interfaces) connect the layer 2 data plane

Physical ports on the front panel of the NFX350 device can be mapped to layer 2 or layer 3 interfaces or VNFs. There is a dedicated IPsec VPN interface (FPC1).

The JCP and PFE perform their primary tasks independently, while constantly communicating with each other. This arrangement provides streamlined forwarding and routing control and the capability to run Internet-scale networks at high speeds.

NFX350 supports numerous routing standards for flexibility and scalability as well as IETF IPSec protocols. These functions can all be managed through the Junos OS software, either from a connected console on the management interface or via a network connection. Network management can be secured using IPsec and SSH protocols. All management, whether from a user connecting to a console or from the network, requires successful authentication. **In the evaluated deployment network management (using the CLI) is secured using the SSH protocol, which can be tunnelled over IPsec.**

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.

In the evaluated configuration the TOE is managed and configured via CLI either via a directly connected console or using SSH connections (optionally tunnelled over IPsec).

1.3.6 Physical Boundaries

The physical boundaries of the TOE is the NFX350 series hardware running Junos OS 20.3R3.

The Junos OS 20.3R3 for NFX350 software includes the KVM Hypervisor as well as the JCP and JDM. Hence the TOE is contained within the physical boundary of each server specified above. The TOE is delivered as a single device with the Junos OS software installed. The TOE model number can be verified through the shipping label and device front panel. The software version can be verified by the show version command once the device is configured.

The Management platform and external syslog server are outside the boundary of the TOE.

1.3.7 Logical Boundary

The logical boundary of the TOE includes the following security functionality:

Table 3 – TOE Logical Boundary Security Functionality

Protected Communications	The TOE provides an SSH server to support protected communications for administrators to establish secure sessions and to connect to external syslog servers. The TOE also supports IPsec connections to provide multi-site virtual private network (VPN) gateway functionality and also as a tunnel for remote administrate SSH connections. The TOE requires that applications exchanging information with it are successfully authenticated prior to any exchange (i.e. applications connecting over SSH and IPsec). Telnet, File Transfer Protocol (FTP), and Secure Socket Layer (SSL) are out of scope. The TOE includes cryptographic modules that provide the underlying cryptographic services, including key management and protection of stored keys, algorithms, random bit generation and crypto-administration. The cryptographic modules provide confidentiality and integrity services for authentication and for protecting communications with adjacent systems.
Administrator Authentication	Administrative users must provide unique identification and authentication data before any administrative access to the system is granted. Authentication data entered and stored on the TOE is protected. The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.
Correct Operation	The TOE provides for both cryptographic and non-cryptographic self-tests and is capable of automated recovery from failure states.
Trusted Update	The administrator can initiate update of the TOE software. The integrity of any software updates is verified prior to installation of the updated software.
Audit	TOE auditable events are stored in the syslog **files in the VM filesystem** and can be sent to an external log server (via Netconf over SSH). Auditable events include start-up and shutdown of the audit functions, authentication events, service requests, IPS events, as well as the events listed in Table 12 and Table 13. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local (VM) syslog storage limits are configurable and are monitored. In the event of storage limits being reached the oldest logs will be overwritten.
Management	The TOE provides a Security 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; • initiation of trusted update function;

	• administration of VPN, IPS and Firewall functionality; • all administrative tasks (e.g., creating the security policy). The devices are managed through a Command Line Interface (CLI). The CLI is accessible through local (serial) console connection or remote administrative (SSH) session. The Security Administrator role includes the capability to manage all NFX350 services. Access to manage the device's FreeBSD host can only be gained through the JCP.
Packet Filtering/Stateful Traffic 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.
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 **using Virtual Routers**. 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).

1.3.8 Non-TOE hardware/software/firmware

The TOE relies on the provision of the following items in the network environment:

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

1.3.9 Security Functions Provided by the TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v2.2e or NDcPP, the collaborative Protection Profile Module for Stateful Traffic Filter Firewalls (MOD_CPP_FW_V1.4E), the PP-Module for Virtual Private Network Gateways (MOD_VPNGW_V1.1) and the PP-Module for Intrusion Prevention Systems (MOD_IPS_V1.0).

1.3.1.3 Security Audit

TOE stored auditable event files locally but can be sent to an external log server (over SSH via Netconf). The TOE provides appropriate start-up and shut-down functions, authentication events, service requests, IPS events and as well events listed in Table 12 and Table 13 of the ST.

Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local (VM) syslog storage limits are configurable and are monitored. In the event of storage limits being reached the oldest logs will be overwritten.

1.3.1.4 Cryptographic Support

The TOE includes cryptographic modules that provide the underlying cryptographic services, including key management and protection of stored keys, algorithms, random bit generation and crypto-administration. The cryptographic modules provide confidentiality and integrity services for authentication and for protecting communications with adjacent systems.

1.3.1.5 Identification and Authentication

Administrative users must provide unique identification and authentication data before any administrative access to the system is granted. Authentication data entered and stored on the TOE is protected. The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.

1.3.1.6 Security Management

Administrative users must provide unique identification and authentication data before any administrative access to the system is granted. Authentication data entered and stored on the TOE is protected. The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.

The TOE provides a Security 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;
- initiation of trusted update function;
- administration of VPN, IPS and Firewall functionality;
- all administrative tasks (e.g., creating the security policy).

The devices are managed through a CLI. The CLI is accessible through local (serial) console connection or remote administrative (SSH) session. The Security Administrator role includes the capability to manage all NFX350 services.

1.3.1.7 Protection of the TSF

The TOE prevents reading of all pre-shared keys, symmetric keys and private keys. Its administrative passwords are store in non-plaintext form to prevent the reading of them. The TOE also runs a suite of self-tests during initial start-up which include the following:

- Power on test
- File Integrity test
- Crypto integrity test
- Authentication test
- Algorithm know answer tests

These self-tests are executed through a TSF provided cryptographic service specified in FCS_COP.1/SigGen.

1.3.1.8 TOE Access

The TOE provides capabilities for controlling session termination, locking and banner display.

1.3.1.9 Trusted Path/Channels

The TOE provides an SSH server to support protected communications for administrators to establish secure sessions and to connect to external syslog servers.

1.3.1.10 Firewall

The TOE uses Stateful Traffic Filtering on network packets that are processed which can permit or drop the capability to the log operations. These Stateful Traffic Filtering rules (found under FFW_RUL.EXT.1) are assigned to a distinct network interface.

1.3.1.11 VPN

The TOE also supports IPsec connections to provide multi-site VPN gateway functionality and also as a tunnel for remote administrative SSH connections. The TOE requires that applications exchanging information with it are successfully authenticated prior to any exchange (i.e. applications connecting over SSH and IPsec).

1.3.1.12 IPS

The TOE includes functionality for policy rules such as white listing via packet monitoring and traffic detection. The TOE performs IP-based analysis of the network traffic in order to detect unusual activity not defined by the administrator. A list of known-good and known-bad IP addresses by source and destination are kept. Additionally a list of known-good and known-bad rules following IPS policy elements is also maintained.

1.3.10 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- [ECG] Common Criteria Configuration Guide for NFX350 Network Services Platform Release 20.3R3
    - Guidance documentation is available online at *https://www.juniper.net/documentation/*

1.4 TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 4 – Required Environmental Components

Components	Description
Management System	The management system is used by an admin to establish a connection to an SSH client used to configure the TOE.
Audit Server	The audit server supports an SSH client which is the trusted channel between itself and the authorized IT entities.
VPN Peer	The VPN peer is a dedicated IPsec interface that established a connection with the VPN gateway.

1.5 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

- Use of telnet, since it violates the Trusted Path requirement set
- Use of FTP, since it violates the Trusted Path requirement set

- Use of SNMP, since it violates the Trusted Path requirement set
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set
- Use of CLI account super-user and linux root account.
- Hosting additional VMs on the TOE physical platform.
- Use of routing protocols such as OSPF, BGP and RIP.

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rational, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017
- Part 2 extended; Part 3 conformant

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- PP-Configuration for Network Device, Intrusion Prevention Systems (IPS), Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, Version 1.0, dated 18 May, 2021 (CFG_NDcPP-IPS-FW-VPNGW_v1.0). This PP-Configuration includes the following components:
  - Base PP: collaborative Protection Profile for Network Devices, version 2.2e, dated 27 March 2020 (CPP_ND_V2.2E),
  - PP-Module: collaborative Protection Profile Module for Stateful Traffic Filter Firewalls, Version 1.4e, dated 01 July 2020 (MOD_CPP_FW_V1.4E),
  - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways, version 1.1, dated 01 July 2020 (MOD_VPNGW_V1.1).
  - PP-Module: PP-Module for Intrusion Prevention Systems (IPS), Version 1.0, dated 11 May 2021 (MOD_IPS_V1.0).

2.3 Conformance Rationale

This ST provides exact conformance to pp-configuration CFG_NDcPP-IPS-FW-VPNGW_V1.0. The security problem definition, security objectives, and security requirements in this ST are all taken from the Protection Profile (PP), modules and packages listed in section 2.2 of Protection Profile Conformance and perform only the operations defined there.

1.3.11 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v2.2e, Firewall Module v1.4e, VPN Gateway Module v1.1, and IPS Module v1.0 have been considered. Table 5 identifies all applicable TDs.

Table 5 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	NTP is not claimed.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
TD0538: NIT Technical Decision for Outdated link to allowed-with list	Yes	
TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63	No	TLS is not claimed.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	FCS_TLS*_EXT.x functionality not claimed.
TD0556: NIT Technical Decision for RFC 5077 question	No	FCS_TLS*_EXT.x functionality not claimed.
TD0563: NiT Technical Decision for Clarification of audit date information	Yes	
TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	FCS_DTLSS_EXT.x and FCS_TLSS_EXT.x functionality not claimed.
TD0570: NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No	TOE is not a virtual TOE.
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD0545: NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	Yes	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0551: NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	Yes	
TD0595: Administrative corrections to IPS PP-Module	Yes	
TD0549: Consistency of Security Problem Definition update for MOD_VPNGW_v1.0 and MOD_VPNGW_v1.1	Yes	
TD0590: Mapping of operational environment objectives	Yes	
TD0597: VPN GW IPv6 Protocol Support	Yes	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	No	TOE is not a virtual Network Device.
TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
TD0634: NIT Technical Decision for Clarification required for testing IPv6	No	FCS_DTLSC_EXT.x and FCS_TLSC_EXT.x functionality not claimed.
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	No	FCS_TLSS_EXT.x functionality not claimed.
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	FCS_SSHC_EXT.x functionality not claimed.

3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP and any relevant EPs/Modules/Packages specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 Threats

The threats included in Table 6 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

Table 6 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of

ID	Threat
	confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.NETWORK_DISCLOSURE (FFW)	An attacker may attempt to "map" a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_ACCESS (FFW)	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_MISUSE (FFW)	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.

ID	Threat
T.MALICIOUS TRAFFIC (FFW)	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.
T.NETWORK_DISCLOSURE (IPS)	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T.NETWORK_ACCESS (IPS)	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. If malicious external devices are able to communicate with devices on the protected network via a backdoor then those devices may be susceptible to the unauthorized disclosure of information.
T.NETWORK_MISUSE (IPS)	Access to services made available by a protected network might be used counter to Operational Environment policies. Devices located outside the protected network may attempt to conduct inappropriate activities while communicating with allowed public services. E.g. manipulation of resident tools, SQL injection, phishing, forced resets, malicious zip files, disguised executables, privilege escalation tools and botnets.
T.NETWORK_DOS (IPS)	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. Resource exhaustion may occur in the event of co-ordinate service request flooding from a small number of sources .
T.DATA INTEGRITY (VPN)	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained within the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS (VPN)	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended

ID	Threat
	for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.
T.NETWORK_ACCESS (VPN)	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.
T.NETWORK_DISCLOSURE (VPN)	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected

ID	Threat
	network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information. From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be 8 prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses and/or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.
T.NETWORK_MISUSE (VPN)	Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services. From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a

ID	Threat
	protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.
T.REPLAY_ATTACK (VPN)	If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a "replay" attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions: Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome. No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these.

3.2 Assumptions

The assumptions included in Table 7 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 7 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

ID	Assumption
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). (NOTE: following paragraph is for virtual network devices. Please delete if the TOE is not a virtual device) In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.
A.NO_THRU_TRAFFIC_PROTECTION[3]	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. (The paragraph that follows is for x509v3 cert-based authentication. If not relevant, remove) For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).

[3] A.NO_THRU_TRAFFIC_PROTECTION is still operative, but only for the interfaces in the TOE that are defined by the Base-PP and not the PP-Module.

ID	Assumption
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
A.CONNECTIONS (IPS)	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.
A.CONNECTIONS (VPN)	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.

3.3 Organizational Security Policies

The OSPs included in Table 8 are drawn directly from the PP and any relevant EPs/Modules/Packages.

Table 8 – OSPs

ID	OSP
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.
P.ANALYZE (IPS)	Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

4 Security Objectives

The security objectives have been taken directly from the claimed PP and any relevant EPs/Modules/Packages and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The security objectives in the following table apply to the TOE.

Table 9 – Security Objectives

ID	Security Objectives
O.RESIDUAL_INFORMATION (FFW)	The TOE shall implement measures to ensure that any previous information content of network packets sent through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both
O.STATEFUL_TRAFFIC_FILTERING (FFW)	The TOE shall perform stateful traffic filtering on network packets that it processes. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified. Depending on the implementation, the TOE might support the stateful traffic filtering of Dynamic Protocols (optional).
O.SYSTEM_MONITORING (IPS)	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.IPS_ANALYZE (IPS)	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.IPS_REACT (IPS)	The IPS must respond appropriately to its analytical conclusions about IPS policy violations.
O.TOE_ADMINISTRATION (IPS)	The IPS will provide a method for authorized administrator to configure the TSF.
O.TRUSTED_COMMUNICATIONS (IPS)	The IPS will ensure that communications between distributed components of the TOE are not subject to unauthorized modification or disclosure.
O.ADDRESS_FILTERING (VPN)	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement Packet Filtering capability. That

ID	Security Objectives
	capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information.
O.AUTHENTICATION (VPN)	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_FUNCTIONS (VPN)	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.FAIL_SECURE (VPN)	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.
O.PORT_FILTERING (VPN)	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.
O.SYSTEM_MONITORING (VPN)	To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only

ID	Security Objectives
	between peer VPN gateways, but also with certification authorities (CAs).
O.TOE_ADMINISTRATION (VPN)	TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.

4.2 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 10 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
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. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION[4]	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

[4] OE.NO_THRU_TRAFFIC_PROTECTION is still operative, but only for the interfaces in the TOE that are defined by the Base-PP and not the PP-Module.

ID	Objectives for the Operational Environment
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
OE.CONNECTIONS (IPS)	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 of monitored networks.
OE.CONNECTIONS (VPN)	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.

5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, September 2017, and all international interpretations.

Table 11 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.1/IPS	Audit Data Generation (IPS)
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.1/IKE	Cryptographic Key Generation (for IKE Peer Authentication)
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_IPSEC_EXT.1	IPsec Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FIA_PSK_EXT.1	Pre-Shared Key Composition
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1/FFW	Specification of Management Functions (FFW)
FMT_SMF.1	Specification of Management Functions

Requirement	Description
FMT_SMF.1/IPS	Specification of Management Functions (IPS)
FMT_SMF.1/VPN	Specification of Management Functions (VPN Gateway)
FMT_SMR.2	Restrictions on security roles
FPF_RUL_EXT.1	Rules for Packet Filtering
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TST_EXT.3	Self-Tests with Defined Methods
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TUD_EXT.1	Trusted Update
FPT_FLS.1/SelfTest	Fail Secure (Self-Test Failures)
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)
FTP_TRP.1/Admin	Trusted Path
FDP_RIP.2	Full Residual Information Protection
FFW_RUL_EXT.1	Stateful Traffic Filtering
FFW_RUL_EXT.2	Stateful Filtering of Dynamic Protocols
IPS_ABD_EXT.1	Anomaly-Based IPS Functionality
IPS_IPB_EXT.1	IP Blocking
IPS_NTA_EXT.1	Network Traffic Analysis
IPS_SBD_EXT.1	Signature-Based IPS Functionality

5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with <u>underlined</u> text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP and relevant EPs/Modules/Packages, the formatting used in the PP has been retained.
- Extended SFRs are identified by the addition of “EXT” after the requirement name.

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) Auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
   - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
   - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
   - *Resetting passwords (name of related user account shall be logged).*
   - [*[Starting and stopping services]*];
d) *Specifically defined auditable events listed in Table 12*.

ST Application Note:

The "Services" referenced in the above requirement relate to the trusted communication channel to the external syslog server (netconf over SSH) and the trusted path for remote administrative sessions (SSH, which can be tunneled over IPsec).

FAU_GEN.1.2

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* Table 12.

Table 12 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.1/IPS	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/SigGen	None	None
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_IPSEC_EXT.1 (VPN)	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
FCS_RBG_EXT.1	None	None
FDP_RIP.2	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
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
FFW_RUL_EXT.2	Dynamical definition of rule Establishment of a session	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None	None
FIA_PSK_EXT.1	None	None
FIA_UAU.7	None	None
FIA_UAU_EXT.2	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MOF.1/Services	None	None
FMT_MTD.1/CoreData	All management activities of TSF data	None
FMT_MTD.1/CryptoKeys	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1	All management activities of TSF data	None
FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules.	None
FMT_SMR.2	None	None
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol
FPT_APW_EXT.1	None	None
FPT_FLS.1/SelfTest	Failure of the TSF	Type of failure that occurred.
FPT_SKP_EXT.1	None	None
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None.	None
FPT_TST_EXT.3	Indication that the TSF self-test was completed Failure of self-test	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None
FTA_SSL.4	The termination of an interactive session	None
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism	None
FTA_TAB.1	None	None

Requirement	Auditable Events	Additional Audit Record Contents
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/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	None

5.2.1.2 FAU_GEN.1/IPS: Audit Data Generation (IPS)

FAU_GEN.1.1/IPS Refinement

The TSF shall be able to generate an **IPS** audit record of the following auditable **IPS** events:

a) Start-up and shut-down 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*.

h) *[no other auditable events]*]

FAU_GEN.1.2/IPS 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~~, [*Specifically defined auditable events listed in Table 12*].

Table 13 – Security Functional Requirements and IPS Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1/IPS	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).

Requirement	Auditable Events	Additional Audit Record Contents
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).
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_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. • The IPS policy and interface mode (if applicable).
IPS_SBD_EXT.1	Inspected traffic matches a signature-based IPS rule with logging enabled.	• 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).

5.2.1.3 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.

5.2.1.4 FAU_STG_EXT.1 Protected Audit Event 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 according to FTP_ITC.1.

ST Application Note
Transfer of the audit data to the external server is performed automatically (without further Security Administrator intervention) in the evaluated deployment.

FAU_STG_EXT.1.2
The TSF Shall be able to store generated audit data on the TOE itself. In addition [*the TOE shall consist of a single standalone component that stores audit data locally*].

FAU_STG_EXT.1.3
The TSF shall [overwrite previous audit records according to the following rule: *[oldest log is overwritten]*] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1
The TSF shall generate **asymmetric** cryptographic key in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].

]

5.2.2.2 FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)

FCS_CKM.1.1/IKE
The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a specified cryptographic key generation algorithm: **[**

- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes;**

and [

- **FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]**

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

5.2.2.3 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Using Discrete Logarithm Cryptography" and [RFC 3526].

5.2.2.4 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a [destruction of reference to the key directly followed by a request for garbage collection];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that *[*
    - instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: No Standard

5.2.2.5 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption
The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CBC, GCM] and [CTR] mode and cryptographic key sizes [128 bits, 256 bits], and [192 bits] that meet the following: AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, GCM as specified in ISO 19772] and [CTR as specified in ISO 10116].

5.2.2.6 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits or 4096 bits*]
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes* [256-bits, 384-bits, 512-bits]

]
that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4

].

5.2.2.7 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

FCS_COP.1.1/Hash
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: *ISO/IEC 10118-3:2004*.

5.2.2.8 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes *[160, 256, 384, and 512 bits]* **and message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.9 FCS_IPSEC_EXT.1 IPSec Protocol

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 have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

FCS_IPSEC_EXT.1.3
The TSF shall implement [tunnel mode].

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 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-256 (RFC 4106)*] and [*AES-CBC-192 (RFC 3602)*] together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-256*]*.

FCS_IPSEC_EXT.1.5
The TSF shall implement the protocol: [

- IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence], and [RFC 4868 for hashfunctions];
- IKEv2 as defined in RFC 5996 and [with no support for NAT traversal], 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-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 5282)].

FCS_IPSEC_EXT.1.7
The TSF shall ensure that [

- IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on
  [
  - length of time, where the time values can be configured within [0.05-24] hours;

  ];
- IKEv2 SA lifetimes can be configured by a Security Administrator based on
  [
  - length of time, where the time values can be configured within [0.05-24] hours

  ]

].

FCS_IPSEC_EXT.1.8
The TSF shall ensure that [

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on
  [
  - number of bytes;
  - length of time, where the time values can be configured within [0.05-8] hours;

  ];
- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on
  [
  - number of bytes;
  - length of time, where the time values can be configured within [0.05-8] hours;

    ]

].

FCS_IPSEC_EXT.1.9
The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least *[112 (for DH Group 14), 128 (for DH Group 19) or 192 (for DH Group 20)]* bits.

FCS_IPSEC_EXT.1.10
The TSF shall generate nonces used in [IKEv1, IKEv2] exchanges of length [

- according to the security strength associated with the negotiated Diffie-Hellman group;
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

  ].

FCS_IPSEC_EXT.1.11
The TSF shall ensure that IKE protocols implement DH Group(s) [

- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and
- [14 (2048-bit MODP)] according to RFC 3526,
- [no other DH groups] according to RFC 5114]

].

FCS_IPSEC_EXT.1.12
The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv1 Phase 1, IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv1 Phase 2, IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13
The TSF shall ensure that all IKE protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys].

FCS_IPSEC_EXT.1.14
The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: Distinguished Name (DN), [SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN].

5.2.2.10 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [HMAC_DRBG (any)*]*.

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[3]* software-based noise source, *[2]* platform-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.2.2.11 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4344, 5656, 6668].

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

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

FCS_SSHS_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr].

FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

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

FCS_SSHS_EXT.1.8
The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[an Administrator configurable positive integer within [1 to 10]]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

5.2.3.2 FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, *[and all other standard ASCII, extended ASCII and Unicode characters]*].
b) Minimum password length shall be configurable to between *[10]* and *[20]* characters.

5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

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;
- [*ICMP echo*].

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

5.2.3.4 FIA_UAU_EXT.1 Password-based Authentication Mechanism

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

5.2.3.5 FIA_UAU.7.1 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**.

5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev
The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates** .
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [no protocols] and [no additional uses].

FIA_X509_EXT.2.2
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [allow the Administrator to choose whether to accept the certificate in these cases].

5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.3.9 FIA_PSK_EXT.1.1 Pre-Shared Key Composition

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

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

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

ST Application Note:

The TOE accepts Unicode characters to specify text-based pre-shared keys. Unicode characters are encoded as UTF-8 and treated as multiple bytes – up to 4 bytes depending on the character. The maximum length limit for text-based pre-shared keys enforced by the TOE is 255 bytes. When a pre-shared key is only composed of ASCII characters this limit is equivalent to 255 characters.

FIA_PSK_EXT.1.3
The TSF shall condition the text-based pre-shared keys by using [SHA1*, [conversion of the text string into an authentication value as per RFC 2409 for IKE1 or RFC 4306 for IKEv2 using the pseudo-random function that is configured as the hash algorithm for the IKE exchanges]*].

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

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity, handling of audit data] to *Security Administrators*.

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the function *to perform manual updates to Security Administrators*.

5.2.4.3 FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services
The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.2.4.4 FMT_MTD.1/CoreData Management of TSF Data

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

5.2.4.5 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to [[manage]] the [cryptographic keys and certificates used for VPN operation] to [Security Administrators].

5.2.4.6 FMT_SMF.1/IPS Specification of Management Functions (IPS)

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

- *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:*
    - *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) signatures*
- *Create custom 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*]

5.2.4.7 FMT_SMF.1.1/VPN Specification of Management Functions (VPN Gateway)

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

- Ability to administer the TOE locally and remotely;
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using digital signature and [no other] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- *Ability to configure all security management functions identified in [VPNGW_MOD];*
- *Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in [VPNGW_MOD];*
- ***Ability to manage the cryptographic keys;***
- ***Ability to configure the cryptographic functionality;***
- ***Ability to configure the lifetime for IPsec SAs;***
- ***Ability to import X.509v3 certificates to the TOE's trust store****;*
- *Definition of packet filtering rules;*
- *Association of packet filtering rules to network interfaces;*
- *Ordering of packet filtering rules by priority;*

[

- *Ability to start and stop services;*
- *Ability to configure audit behavior(e.g. changes to storage locations for audit; changes to behavior when local audit storage space is full);*
- *Ability to configure thresholds for SSH rekeying;*
- *Ability to re-enable an Administrator account;*
- *Ability to set the time which is used for time-stamps;*
- *Ability to configure the reference identifier for the peer;*
- *Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;*
- *Ability to manage the trusted public keys database;*

].

5.2.4.8 FMT_SMF.1/FFW Specification of Management Functions

FMT_SMF.1/FFW
The TSF shall be capable of performing the following functions:

- *Ability to configure firewall rules;*

5.2.4.9 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1
The TSF shall maintain the roles:

- *Security 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

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

5.2.5 Packet Filtering (FPF)

5.2.5.1 FPF_RUL_EXT.1.1 Rules for Packet Filtering

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

FPF_RUL_EXT.1.2
The TSF shall allow the definition of Packet Filtering rules using the following network protocols and protocol fields:

- IPv4 (RFC 791)
  - Source address
  - Destination Address
  - Protocol
- IPv6 (RFC 2460)
  - Source address
  - Destination Address
  - Next Header (Protocol)

- TCP (RFC 793)
  - Source Port
  - Destination Port
- UDP (RFC 768)
  - Source Port
  - Destination Port

FPF_RUL_EXT.1.3
The TSF shall allow the following operations to be associated with Packet Filtering rules: permit and drop with the capability to log the operation.

FPF_RUL_EXT.1.4
The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.5
The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

FPF_RUL_EXT.1.6
The TSF shall drop traffic if a matching rule is not identified.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FTP_APW_EXT.1 Protection of Administrator Passwords

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

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

5.2.6.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric, and private keys)

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

5.2.6.3 FPT_STM_EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2
The TSF shall [allow the Security Administrator to set the time].

5.2.6.4 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [during initial startup (on power on)] to demonstrate the correct operation of the TSF: **noise source health tests,** [

- *Power on test,*
- *File integrity test,*
- *Crypto integrity test,*

- *Authentication test,*
- *Algorithm known answer tests*].

5.2.6.5 FPT_TST_EXT.3 TSF Self-Test with Defined Methods

FPT_TST_EXT.3.1
The TSF shall run a suite of the following self-tests [*[when loaded for execution]*] to demonstrate the correct operation of the TSF: [*integrity verification of stored executable code*].

FPT_TST_EXT.3.2
The TSF shall execute the self-testing through [*a TSF-provided cryptographic service specified in FCS_COP.1/SigGen*].

5.2.6.6 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1
The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD_EXT.1.2
The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3

The TSF shall provide the means to authenticate firmware/software updates to the TOE using a *digital signature mechanism* and *[no other mechanisms]* prior to installing those updates.

5.2.6.7 FPT_FLS.1/SelfTest Fail Secure

FPT_FLS.1.1/SelfTest
The TSF shall **shut down** when the following types of failures occur: [failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests].

5.2.7 TOE Access (FTA)

5.2.7.1 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 period of inactivity.

5.2.7.2 FTA_SSL.3 TSF-initiated Termination

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

5.2.7.3 FTA_SSL.4 User-initiated Termination

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

5.2.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1
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.

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

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

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

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[none]*.

5.2.8.2 FTP_ITC.1/VPN Inter-TSF Trusted Channel (VPN Communications)

FTP_ITC.1.1/VPN
The TSF shall **be capable of using IPsec to** provide a communication channel between itself and **authorized IT entities supporting VPN communications** 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/VPN
The TSF shall permit [*the authorized IT entities*] to initiate communication via the trusted channel.

FTP_ITC.1.3/VPN
The TSF shall initiate communication via the trusted channel for [remote VPN gateways/peers].

5.2.8.3 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [SSH] to** provide a communication path between itself and **authorized** 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 provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

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

5.2.9 User Data Protection (FDP)

5.2.9.1 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.

5.2.10 Firewall (FFW)

5.2.10.1 FFW_RUL_EXT.1 Stateful Traffic 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 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*
  - *[no other field]*
- *TCP*
  - *Source Port*
  - *Destination Port*
- *UDP*
  - *Source Port*
  - *Destination Port*

and distinct interface.

FFW_RUL_EXT.1.3
The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4
The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5
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]]*.

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.1.6

The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:

a) The TSF shall drop and be capable of [logging] packets which are invalid fragments;
b) The TSF shall drop and be capable of [logging] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall drop 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" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
i) [no other rules].

FFW_RUL_EXT.1.7

The TSF shall be capable of dropping and logging according to the following rules:

a) The TSF shall drop 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;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) The TSF shall drop 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.

FFW_RUL_EXT.1.8

The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

FFW_RUL_EXT.1.9

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

FFW_RUL_EXT.1.10
The TSF shall be capable of limiting an administratively defined number of *half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [logged].*

5.2.10.2 FFW_RUL_EXT.2 Stateful Filtering of Dynamic Protocols

FFW_RUL_EXT.2.1
The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [FTP].

5.2.11 Intrusion Prevention (IPS)

5.2.11.1 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;
- frequency;
- thresholds;
- [*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
  - send a TCP reset to the source address of the offending traffic;
  - send a TCP reset to the destination address of the offending traffic;
- In inline mode:
  - allow the traffic flow
  - block/drop the traffic flow
  - and [no other actions]

5.2.11.2 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 and [no additional address types].

IPS_IPB_EXT.1.2
The TSF shall allow [Security Administrators] to configure the following IPS policy elements: [known-good list rules, known-bad list rules, IP addresses, no other IPS policy elements].

5.2.11.3 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: [FastEthernet interface: dedicated management Ethernet interface];
    - [Session-reset-capable interfaces: [*Ethernet interfaces*];
    - no other interface types].

5.2.11.4 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 to 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; IP Options; and [no other field].
- IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and [traffic class, flow label].
- 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.
  iv) [*no other types of TCP payload inspection*]];
- UDP data: characters beyond the first 8 bytes of the UDP header;

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
   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;
  - send a TCP reset to the source address of the offending traffic;
  - send a TCP reset to the destination address of the offending traffic;]
- In inline mode:
  - block/drop the traffic flow;
  - and [
    - allow all traffic flow;]

IPS_SBD_EXT.1.6
The TSF shall support stream reassembly or equivalent to detect malicious payload even if it is split across multiple non-fragmented packets

5.3 TOE SFR Dependencies Rationale for SFRs

The PP and any relevant EPs/Modules/Packages contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the Table 14.

Table 14 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Juniper to satisfy the assurance requirements. The following table lists the details.

Table 15 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

6 TOE Summary Specifications

This chapter identifies and describes how the Security Functional Requirements identifies above are met by the TOE.

The following table relates cryptographic algorithms to the protocols implemented in the TOE. The TOE acts as both sender and recipient for IPsec and only as the server for SSH in the supported protocols listed in Table 16:

Table 16 – Protocol Usage of Cryptographic Algorithms

Protocol	Key Exchange	Auth	Cipher	Integrity
IKEv1	Group 14 (modp 2048) Group 19 (P-256) Group 20 (P-384)	RSA 2048 ECDSA P-256 ECDSA P-384 Pre-Shared Key	AES CBC 128 AES-CBC-192 AES CBC 256	HMAC-SHA-256-128 HMAC-SHA-384-192
IKEv2	Group 14 (modp 2048) Group 19 (P-256) Group 20 (P-384)	RSA 2048 ECDSA P-256 ECDSA P-384 Pre-Shared Key	AES CBC 128 AES-CBC-192 AES CBC 256 AES GCM 128 AES GCM 192 AES GCM 256	HMAC-SHA-256-128 HMAC-SHA-384-192
IPsec ESP	IKEv1 with optional: • Group 14 (modp 2048) • Group 19 (P-256) • Group 20 (P-384)	IKEv1	AES CBC 128 AES-CBC-192 AES CBC 256	HMAC-SHA-256-128
	IKEv2 with optional: • Group 14 (modp 2048) • Group 19 (P-256) • Group 20 (P-384)	IKEv2	AES CBC 128 AES-CBC-192 AES CBC 256 AES GCM 128 AES-GCM-192 AES GCM 256	HMAC-SHA-256-128
SSHv2	Diffie-Hellman Group 14 (modp 2048) ECDH-sha2-nistp256 ECDH-sha2-nistp384 ECDH-sha2-nistp521	ECDSA P-256 ECDSA P-384 ECDSA P-521	AES CTR 128 AES CTR 256 AES CBC 128 AES CBC 256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

Table 17 – TOE Summary Specification SFR Description

Requirement	TSS Description
FAU_GEN.1, FAU_GEN.1/IPS, FAU_GEN.2, FAU_GEN_EXT.1	Junos OS creates and stores audit records for the following events (the detail of content recorded for each audit event is detailed in Table 12 and Table 13. Auditing is implemented using syslog. • Start-up and shut-down of the audit functions • Administrative login and logout

Requirement	TSS Description
	• Configuration is committed • Configuration is changed (includes all management activities of TSF data) • Generating/import of, changing, or deleting of cryptographic keys (see below for more detail) • Resetting passwords • Starting and stopping services • All use of the identification and authentication mechanisms • Unsuccessful login attempts limit is met or exceeded • Any attempt to initiate a manual update • Result of the update attempt (success or failure) • The termination of a local/remote/interactive session by the session locking mechanism • Initiation/termination/failure of the SSH trusted channel to syslog server • Initiation/termination/failure of the SSH trusted path with Admin • Initiation/termination/failure of an IPsec trusted channel, including Session Establishment with peer • Session establishment with CA • 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 • 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 and shut-down 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 with logging enabled • Inspected traffic matches an anomaly-based IPS policy In addition the following management activities of TSF data are recorded: • configure the access banner; • configure the session inactivity time before session termination; • configure the authentication failure parameters for FIA_AFL.1; • Ability to configure audit behaviour; • configure the cryptographic functionality; • configure thresholds for SSH rekeying; • re-enable an Administrator account; • set the time which is used for time-stamps.

Requirement	TSS Description
	The detail of what events are to be recorded by syslog are determined by the logging level specified the "level" argument of the "`set system syslog`" CLI command. To ensure compliance with the requirements the audit knobs must be configured. As a minimum, Junos OS 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). Because of the nature of IPS event logs, log generation often happens in bursts and can generate a large 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. IPS log suppression is enabled by default and can be customized based on the following 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 entries containing the count of occurrences. Traffic 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. In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys): • PKID – certificate id will be recorded when generating or deleting a key pair • IKE SPI – IP address of the initiator and responder recorded, together with the SPI, will be recorded when generating a key pair. The IP address of the initiator and responder will provide the unique link to the key identifier (SPI) of the key that has been destroyed in the session termination • SSH session keys– key reference provided by process id • SSH key configured for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog For SSH (ephemeral) session keys the PID is used as the key reference to relate the key generation and key destruction audit events. The key destruction event is recorded as a session disconnect event. For example, key generation and key destruction events for a single SSH session key would be reflected by records similar to the following: Sep 27 15:09:36 yeti sshd[6529]: Accepted publickey for root from 10.163.18.165 port 45336 ssh2: RSA SHA256:l1vri77TPQ4VaupE2NMYiUXPnGkqBWIgD5vW0OuglGI … Sep 27 15:09:40 yeti sshd[6529]: Received disconnect from 10.163.18.165 port 45336:11: disconnected by user

Requirement	TSS Description
	Sep 27 15:09:40 yeti sshd[6529]: Disconnected from 10.163.18.165 port 45336 It should be noted that SSH keys used for trusted channels are NOT deleted by mgd when SSH is de-configured. Hence, the only time SSH keys used for trusted channels are deleted is when a "request system zeroize" action is performed and the whole VM is zeroized (which by definition cannot be recorded). All events recorded by syslog are timestamped. The clock function of Junos OS provides a source of date and time information for the appliance, used in audit timestamps. The clock is also used to determine certificate expiration, administrator session timeouts, and IPsec/SSH rekey thresholds. Wind River Linux kernel provides the current time when it bootstraps the Junos OS VM. Once the Junos OS VM is started it maintains its own time using the hardware Time Stamp Counter as the clock source.
FAU_STG_EXT.1	The TOE is a standalone device wherein syslog can be configured to store the audit logs locally, and optionally to send them to one or more syslog log servers in real time via Netconf over SSH. Local audit log are stored in /var/log/ in the underlying filesystem. Only a Security Administrator can read log files or delete log and archive files through the CLI interface or through direct access to the filesystem having first authenticated as a Security Administrator. The syslogs are automatically deleted locally according to configurable limits on storage volume. The default maximum size is 1Gb. The default maximum size can be modified by the user, using the "`size`" argument for the "`set system syslog`" CLI command. The Junos OS 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. A 1Gb syslog file takes approximately 0.25Gb of storage when archived. Syslog files can acquire complete storage allocated to /var filesystem, which is platform specific. However, when the filesystem reaches 92% storage capacity an event is raised to the administrator but the event process (being a privileged process) still can continue using the reserved storage blocks. This allows the syslog to continue storing events while the administrator frees the storage. If the administrator does not free the storage in time and the /var filesystem storage becomes exhausted a final entry is recorded in the log reporting "No space left on device" and logging is terminated. The appliance continues to operate in the event of exhaustion of audit log storage space.
FCS_CKM.1	The TOE's cryptographic module generates asymmetric keys. The asymmetric keys produced are: • RSA 2048, 4096 bit • ECC (P-256, P-384, P-521) • DH group 14 (2048 bits) Usage of the keys in protocols is specified in Table 16.

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td>FCS_CKM.1/IKE</td><td>Asymmetric keys are generated in accordance with NIST SP 800-56A and FIPS PUB 186-4 for IKE with IPSec. The TOE complies with section 5.6 of NIST SP 800-56A regarding asymmetric 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 186-4 Appendix B.3.3 and B.4.2.
There are no other TOE-specific extensions or processes not included in the Appendices or alternative Implementations allowances that may impact the security requirements.</td></tr>
<tr><td>FCS_CKM.2</td><td>Asymmetric keys are also generated in accordance with FIPS PUB 186-4 Appendix B.3.3 for RSA Schemes and Appendix B.4.2 for ECC Schemes for SSH and IPsec communications. The TOE implements Diffie-Hellman group 14, using the modulus and generator specified by Section 3 of RFC3526. Usage of key agreement in protocols is specified in Table 16.</td></tr>
<tr><td>FCS_CKM.4</td><td>Table 19 of the Security Target lists all relevant keys as well as their origin, storage location, situations in which keys are destroyed and key destruction method used. The TOE stores plaintext keys in volatile and non-volatile memory. Keys listed are consistent with the functions carried out by the TOE. There are no configurations that do not conform to the key destruction requirement.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>Usage of encryption with protocols is specified in Table 16. This information includes modes and key sizes.</td></tr>
<tr><td>FCS_COP.1/Hash</td><td>Hash functions are used in support of protocols as specified in Table 16 and Table 18. SHA-1, SHA-256 and SHA-512 are also used for password hashing.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>The following table states the key lengths, hash functions, block sizes and output MAC lengths supported by the TOE.

<table>
<tr><th>HMAC-SHA</th><th>-1</th><th>-256</th><th>-384</th><th>-512</th></tr>
<tr><td>Key Length</td><td>160 bits</td><td>256 bits</td><td>384 bits</td><td>512 bits</td></tr>
<tr><td>Hash function</td><td>SHA-1</td><td>SHA-256</td><td>SHA-384</td><td>SHA-512</td></tr>
<tr><td>Block Size</td><td>512 bits</td><td>512 bits</td><td>1024 bits</td><td>1024 bits</td></tr>
<tr><td>Output MAC</td><td>160 bits</td><td>256 bits</td><td>384 bits</td><td>512 bits</td></tr>
</table>
</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE performs cryptographic signature services (generation and verification) using the following cryptographic algorithms:
• RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits or 4096 bits]
• Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256-bits, 384-bits, 512-bits]</td></tr>
<tr><td>FCS_IPSEC_EXT.1</td><td>The TOE is conformant to RFC 4301.
The TOE supports tunnel mode only.
By default, the TOE denies all traffic through the NFX350 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.</td></tr>
</table>

Requirement	TSS Description
	The security policy rule set is an ordered list of security policy entries enforced by the firewall rules, 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 and the packet is discarded. When a packet is processed by the TOE, the route is checked to see if it meets a defined security policy. If the packet meets the security policy, it is processed according to the rules of that policy. The following modes can be defined for a security flow policy: • Bypass – The Permit option directs traffic traversing the device through the stateful firewall inspection, but not through the IPsec VPN tunnel. • Discard – The Deny option inspects and drops all packets that do not match any Permit policies. • Protect – The traffic is routed through an IPsec tunnel based on a combination of route lookup and Permit policy inspection. • Log – This option logs traffic and session information for all modes mentioned above. For inbound traffic, the TOE looks up the SA by using the destination IP address, security protocol, and security parameter index (SPI) value. For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel. If a packet arrives and there is not an active SA for that tunnel, the packet is dropped. The TOE will then begin to establish a tunnel, so that when the packet is resent, the SA is active. After the SA is established all subsequent packets in the session will use the IPsec tunnel. The TOE supports AES-GCM-128, and AES-GCM-256, and AES-CBC-128, AES-CBC-192 or AES-CBC-256 using HMAC SHA-256 for ESP protection. 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 no support for NAT traversal) and RFC 4868 for hash functions. IKEv1 aggressive mode is not supported. The TOE supports AES-CBC-128, AES-CBC-192, and AES-CBC-256 for payload protection in IKEv1 and IKEv2. The TOE also supports AES-GCM-128 and AES-GCM-256 for the payload protection in IKEv2. In the evaluated configuration, the TOE permits configuration of the: • IKEv1 Phase 1 and IKEv2 SA lifetimes in terms of length of time (180 to 86400 seconds), • IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes in terms of (kilo)bytes (64 to 1GB) lifetime or length of time (180 to 28,800 seconds) The following CLI commands configure a lifetime of either kilobytes or seconds:

Requirement	TSS Description

	`set security ipsec proposal <name> lifetime-kilobytes <kb>`
	`set security ipsec proposal <name> lifetime-seconds <seconds>`
	The TOE supports Diffie-Hellman Groups 14, 19, and 20. 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, or 20) 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.
	The TOE uses HMAC DRBG with SHA-256 for the generation of DH exponents and nonces in the IKE key exchange protocol of length 224 bits (for DH Group 14).
	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 checks the strengths of the configured IKE algorithms prior to committing a tunnel configuration to ensures that the strength of the symmetric algorithm (128, 192 or 256 bits) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection. If the strength is not greater an error is displayed, and the configuration fails.
	The TOE uses pre-shared keys for IPSec. The TOE accepts Unicode characters to specify text-based pre-shared keys. Unicode characters are encoded as UTF-8 and treated as multiple bytes – up to 4 bytes depending on the character. The maximum length limit for text-based pre-shared keys enforced by the TOE is 255 bytes. When a pre-shared key is only composed of ASCII characters this limit is equivalent to 255 characters. The text-based pre-shared or bit-based keys may contain upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. The TOE accepts pre-shared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges. There is no difference between how the TOE processes text-based or bit-based pre-shared keys.

FCS_RBG_EXT.1	Junos OS performs random bit generation in accordance with NIST Special Publication 800-90 using HMAC_DRBG, SHA-256. The RBG for the NFX does not require any configuration and is seeded with 256 bits of entropy from hardware-based noise sources of entropy, RANDOM_INTERRUPT, RANDOM_ATTACH, as well as a software-based noise source, namely RANDOM_NET_ETHER, RANDOM_SWI, RANDOM_FS_ATIME: • RANDOM_INTERRUPT: This hardware source of entropy is provided by devices whose hardware interrupts are known to provide some amount of entropy, such as hard drive controllers. The timings are fed into kernel HMAC DRBG (Juniper kernel DRBG) along with a CPU cycle counter. This source of provides entropy during the boot-up process and during steady state operations.

Requirement	TSS Description
	• RANDOM_NET_ETHER: This source of entropy is associated with network activity. Timings (CPU counter values at the time of the event) together with internal representation of network packets are used to construct extra entropy that is fed into Kernel HMAC DRBG. This source will only provide entropy after the device has booted and has started processing network packets. • RANDOM_SWI: This source of entropy is associated with software thread interrupts. Timing of software interrupts are combined with event and thread pointers to construct extra entropy that is fed into the Kernel HMAC DRBG. This source of provides entropy during the boot-up process and during steady state operations. • RANDOM_FS_ATIME: This source of entropy is associated with temporary file storage. An internal representation of the file system node of a file in temporary storage is hashed and used to construct entropy that is fed into the Kernel HMAC DRBG. This source of provides entropy during the boot-up process and during steady state operations. • RANDOM_ATTACH: This source of entropy is associated with attaching devices. The timing delta for the time to attach a device is used to construct entropy that is fed into the Kernel HMAC DRBG. This source of entropy provides entropy only during the boot-up process.
FCS_SSHS_EXT.1	Junos OS provides an SSH server to support Trusted Channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The remote audit server initiates the connection. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. The Junos OS SSH Server also supports Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between Junos OS SSH Server and a remote administrator is provided by the use of an SSH session. Remote administrators of Junos OS initiate communication to the Junos CLI through the SSH tunnel created by the SSH session. Assured identification of Junos OS is guaranteed by using public key based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. Junos OS SSH server is implemented in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. Junos OS provides assured identification of the Junos OS appliance though public key authentication and supports password-based authentication by administrative users (Security Administrator) for SSH connections. **RFC 4251:** Host Keys: The TOE uses an ECDSA Host Key for SSHv2, with a default key size of 256 bits, which is generated on initial setup of the TOE. It can be de-configured via the CLI and the key will be deleted and thus unavailable during connection establishment. This key is randomly generated to be unique to each

Requirement	TSS Description
	TOE instance. The TOE presents the client with its public key and the client matches this key against its known_hosts list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different (per the SSHv2 protocol). Policy Issues: The TOE implements all mandatory algorithms and methods. The TOE can be configured to accept public-key based authentication and/or password-based authentication. The TOE does not require multiple authentication mechanisms for users. The TOE allows port forwarding and sessions to clients. The TOE has no X11 libraries or applications and X11 forwarding is prohibited. Confidentiality: The TOE does not accept the “none” cipher and supports AES-CBC-128, AES-CBC-256, AES-CTR-128, AES-CTR-256 encryption algorithms for protection of data over SSH and uses keys generated in accordance with “ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521” to perform public-key based device authentication. For ciphers whose blocksize >= 16, the TOE rekeys every (2^32-1) bytes. The client may explicitly request a rekeying event as a valid SSHv2message at any time and the TOE will honor this request. Re-keying of SSH session keys can be configured using the sshd_config knob. The data-limit must be set between 51200 and 1Gbyte and the time-limit must be set within 1 and 60 minutes. The TOE will rekey based on whichever limit is reached first. Denial of Service: When the SSH connection is brought down, the TOE does not attempt to re-establish it. Ordering of Key Exchange Methods: Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (specified in RFC 5656), diffie-hellman-group14-sha1 (specified in RFC 4253). Debug Messages: The TOE sshd server does not support debug messages via the CLI. End Point Security: The TOE permits port forwarding. Proxy Forwarding: The TOE permits proxy forwarding. X11 Forwarding: The TOE does not support X11 forwarding. **RFC 4252:** Authentication Protocol: The TOE does not accept the “none” authentication method. The TOE implements a timeout period of 30 seconds for authentication of the SSHv2 protocol and provides a limit of three failed authentication attempts before sending a disconnect to the client. Authentication Requests: The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed – it sends back a disconnect message as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies “none” authentication method and replies with a list of permitted authentication methods. Public Key Authentication Method: The TOE supports public key authentication for SSHv2 session authentication using the following algorithms: ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521. Authentication succeeds if the correct private key is used. The TOE does not require multiple authentications (public key and password) for users.

Requirement	TSS Description
	Password Authentication Method: The TOE supports password authentication. Expired passwords are not supported and cannot be used for authentication. Host-Based Authentication: The TOE does not support the configuration of host-based authentication methods. **RFC 4253:** Encryption: The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc, aes128-ctr, aes256-ctr. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the "none" algorithm for encryption. Maximum Packet length: The TOE reads the packet payload size in TCP packets to determine packet length. Packets greater than 256K bytes in an SSH transport connection are dropped and the connection is terminated by Junos OS. Data Integrity: The TOE permits negotiation of HMAC-SHA1 in each direction for SSH transport. Key Exchange: The TOE supports diffie-hellman-group14-sha1. Key Re-Exchange: The TOE performs a re-exchange when SSH_MSG_KEXINIT is received. **RFC 4254:** Multiple channels: The TOE assigns each channel a number (as detailed in RFC 4251, see above). Data transfers: The TOE supports a maximum window size of 256K bytes for data transfer. Interactive sessions: The TOE only supports interactive sessions that do NOT involve X11 forwarding. Forwarded X11 connections: This is not supported in the TOE. Environment variable passing: The TOE only sets variables once the server process has dropped privileges. Starting shells/commands: The TOE supports starting one of shell, application program or command (only one request per channel). These will be run in the context of a channel, and will not halt the execution of the protocol stack. Window dimension change notices: The TOE will accept notifications of changes to the terminal size (dimensions) from the client. Port forwarding: This is fully supported by the TOE. **RFC 4344:** Encryption modes: The TOE uses AES128-ctr and AES256-ctr for encryption. **RFC 5656:** ECDH Key Exchange: The supported key exchange method specified in this RFC are ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521. The client matches the key against its known_hosts list of keys. Hashing: Junos OS supports cryptographic hashing via the SHA-1, SHA-256 and SHA-384 algorithms. Required Curves: curves are implemented: ecdh-sha2-nistp256. None of the Recommended Curves are supported as they are not included in [ND_cPP]. **RFC 6668:**

Requirement	TSS Description
	Data Integrity Algorithms: Both the recommended and optional algorithms hmac-sha2-256 and hmac-sha2-512 (respectively) are implemented for SSH transport.
FDP_RIP.2	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 overwritten with zeros (making the previous data unavailable or zeroized) 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.
FFW_RUL_EXT.1, FFW_RUL_EXT.2, FPF_RUL_EXT.1	The boot sequence of the TOE appliances also aids in establishing the securing domain and preventing tampering or bypass of security functionality. This includes ensuring the packet filtering rules cannot be bypassed during the boot sequence of the TOE. 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 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. 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.

Requirement	TSS Description
	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

Requirement	TSS Description
	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: • RFC 792 ICMPv4: Type, Code • RFC 4443 ICMPv6: Type, Code • RFC 791 (IPv4): Source address, Destination Address, Transport Layer Protocol • RFC 2460 (IPv6): Source address, Destination Address, Transport Layer Protocol • RFC 793 (TCP): Source port, Destination port • RFC 768 (UDP): Source port, Destination port 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 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 Administrator rules. Session events will be logged

Requirement	TSS Description
	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 TOE enforces 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: o No overlap o The total fragments in one packet should not be more than 62 pieces o The total length of merged fragments should not larger than 64k o All fragments in one packet should arrive in 2 seconds o The total queued fragments has limitation, depending on the platform o The total number of concurrent fragment processing for different packet has limitations depending on platform The TOE can be configured to drop connection attempts after a defined number of half-open TCP connections using the Junos screen 'tcp syn-flood', which provides both source and destination thresholds on the number of uncompleted TCP connections, as well as a timeout period. The source

Requirement	TSS Description
	threshold option allows administrators to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address—before Junos OS begins dropping connection requests from that source. Similarly, the destination threshold option allows administrators to specify the number of SYN segments received per second for a single destination IP address before Junos OS begins dropping connection requests to that destination. The timeout option allows administrators to set the maximum length of time before an uncompleted connection is dropped from the queue.
FIA_AFL.1	The retry-options can be configured to specify the action to be taken if the administrator fails to enter valid username/password credentials for password authentication when attempting to authenticate via remote access. The retry-options are applied following the first failed login attempt for a given username. The length of delay (5-10 seconds) after each failed attempt is specified by the backoff-factor, and the increase of the delay for each subsequent failed attempt is specified by the backoff-threshold (1-3). The tries-before-disconnect sets the maximum number of times (1-10) the administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected. Each failed attempt is tracked by the username. When the tries-before-disconnect number is reached for any particular user, that username is locked and cannot be used to authenticate remotely. The lockout-period sets the amount of time in minutes before the administrator can attempt to log in to the device after being locked out due to the number of failed login attempts (1-43,200 minutes). Even when an account is blocked for remote access to the TOE, an administrator is always able to login locally through the serial console and the administrator can attempt authentication via remote access after the maximum timeout period of 24 hours.
FIA_PMG_EXT.1	Authentication data for fixed password authentication is a case-sensitive, alphanumeric value. The password has a: • Minimum length of 10 characters and maximum length of 20 characters • Must contain characters from at least three different character sets (upper, lower, numeric, punctuation) • Can be up to 20 ASCII characters in length (control characters are not recommended). Any standard ASCII, extended ASCII and Unicode characters can be selected when choosing a password.
FIA_PSK_EXT.1	The TOE uses pre-shared keys for IPSec. The TOE accepts Unicode characters to specify text-based pre-shared keys. Unicode characters are encoded as UTF-8 and treated as multiple bytes – up to 4 bytes depending on the character. The maximum length limit for text-based pre-shared keys enforced by the TOE is 255 bytes. When a pre-shared key is only composed of ASCII characters this limit is equivalent to 255 characters. The text-based pre-shared or bit-based keys may contain upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. The TOE accepts pre-shared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges. There is no

Requirement	TSS Description
	difference between how the TOE processes text-based or bit-based pre-shared keys.
FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7	Junos users are configured under "system login user" and are exported to the password database '/var/etc/master.passwd'. A Junos user is therefore an entry in the password database. Each entry in the password database has fields corresponding to the attributes of "system login user", including username, (obfuscated) password and login class. The internal architecture supporting Authentication includes an active process, associated linked libraries and supporting configuration data. The Authentication process and library are: • login() • PAM Library module Following TOE initialization, the 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 login process does two things; it first establishes that the requesting user is whom they claim to be and second provides them with an interactive Junos Command interactive command line interface (CLI). The SSH daemon supports public key authentication by looking up a public key in an authorized keys file located in the directory '.ssh' in the user's home directory (i.e. '~/.ssh/') and this authentication method will be attempted before any other if the client has a key available. The SSH daemon will ignore the authorized keys file if it or the directory '.ssh' or the user's home directory are not owned by the user or are writeable by anyone else. For password authentication, login() interacts with a user to request a username and password to establish and verify the user's identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed. login() uses PAM Library calls for the actual verification of this data. The password is hashed and compared to the stored value, and success/failure is indicated to login(). PAM is used in the TOE to support authentication management, account management, session management and password management. Login primarily uses the session management and password management functionality offered by PAM. The TOE requires users to provide unique identification and authentication data (passwords/public key) before any access to the system is granted. Prior to authentication, the only Junos OS managed responses provided to the administrator are: • Display of the access banner • ICMP echo responses.
FIA_X509_EXT.1/Rev, FIA_X509_EXT.2	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 X.509 certificates as defined in RFC 5280.

Requirement	TSS Description
	When certificates are used for authentication in IPsec, the certificate validity checking is performed anytime the certificate is presented for authentication. 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 using CRL (as specified in RFC 5280 Section 6.3). 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. The TOE validates a certificate path by building a chain of (at least 3) 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 email address, fully qualified domain name or IP address. When configuring an IKE policy, the certificate name must be set so the TOE knows which certificate to use for authentication. If either the certificate does not validate, or the contents do not match the configured identity, then the SA will not be established. When configuring the IKE identity of the remote endpoint the administrator must specify an email address, fully qualified domain name, or IP address that will be matched against the SAN field, or a distinguished name, in the presented certificate. If the TSF cannot establish a connection to determine the validity of a certificate, the TOE takes the action configured by the administrator. In the NDcPP deployment, “disable on-download-failure” may be set for a CA to allow connections to be established when CRLs could not be retrieved . Otherwise, connections involving a CA for which the specified CRL could not be retrieved are rejected. For public key-based authentication of IPsec connections, Junos OS validates the X.509 certificates by extracting the subject, issuer, 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 CA is not present, or if the issuer certificate does not have the CA:true flag in the basicConstraints section, the validation fails. Junos OS 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.

Requirement	TSS Description
FIA_X509_EXT.3	Junos OS generates Certificate Request Messages as specified in RFC 2986. Junos OS validates the chain of certificates from the Root CA when the CA Certificate Response is received. 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
FMT_MOF.1/Functions	Syslog can be configured to store the audit logs locally, and optionally to send them to one or more syslog log servers in real time via Netconf over SSH.
FMT_MOF.1/ManualUpdate	Security Administrators are able to initiate an update of the TOE firmware if a new version of the TOE firmware is available. Updates are downloaded and applied manually (there is no automatic updating of the Junos OS).
FMT_MOF.1/Services	Security Administrators are able to manage the following functions: • Transmission of audit data to an external IT entity, including Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH) • Handling of audit data, including setting limits of log file size
FMT_MTD.1/CoreData	The Administrator is the only role with the ability to manage the TSF data. The Administrator can perform management functions via a specialized interface. No administrative functions are accessible prior to logging in.
FMT_MTD.1/CryptoKeys	The Security Administrator has the capability to: • Manage crypto keys: ○ SSH key generation (ecdsa, ssh-rsa)

FMT_SMF.1/FFW, FMT_SMF.1/IPS, FMT_SMF.1/VPN	The Security Administrator is associated with the defined login class “security-admin”, which has the necessary permission set to permit the administrator to perform all tasks necessary to manage Junos OS in accordance with the requirements of [ND_cPP], using both the local as well as the remote administrative interface.
	The Security Administrator has the capability to:
	• Administer the TOE locally via the serial ports on the physical device or remotely over an SSH connection.
	• Initiate a manual update of TOE software:
	  ○ Query currently executing version of TOE software (both Junos OS and underlying Wind River Linux Host OS)
	  ○ Verify update using published digital signature
	• Manage Functions:

Requirement	TSS Description
	○ Transmission of audit data to an external IT entity, including Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH) ○ Handling of audit data, including setting limits of log file size • Manage TSF data: ○ Create, modify, delete administrator accounts, including configuration of authentication failure parameters ○ Reset administrator passwords • Manage crypto keys: ○ SSH key generation (ecdsa, ssh-rsa) • Perform management functions: ○ Configure the access banner ○ Configure the session inactivity time before session termination or locking, including termination of session when serial console cable is disconnected ○ Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors; ○ Ability to import X.509v3 certificates ○ Manage cryptographic functionality, including: ▪ ssh ciphers ▪ hostkey algorithm ▪ key exchange algorithm ▪ hashed message authentication code ▪ thresholds for SSH rekeying ○ Set the system time ○ Ability to configure Firewall rules; ○ Ability to configure the VPN-associated cryptographic functionality; ○ Definition of packet filtering rules; ○ Association of packet filtering rules to network interfaces; ○ Ordering of packet filtering rules by priority; ○ Ability to configure the IPsec functionality, including configuration of IKE lifetime-seconds (within range 180 to 86400 , with default value of 180 seconds), IPsec lifetime-seconds (within range 180 to 86400, with default value of 28800 seconds), and Lifetime-kilobytes (within range 64 to 4294967294 kilobytes) and ability to configure the reference identifier for the peer; ○ 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 analysed: ▪ Source IP addresses (host address and network address);

Requirement	TSS Description
	▪ 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. Security Administrators are able to initiate an update of the TOE firmware if a new version of the TOE firmware is available. Updates are downloaded and applied manually (there is no automatic updating of the Junos OS).
FMT_SMR.2	Junos OS enforces binding between human users and subjects. The Security Administrator is responsible for provisioning Junos OS user accounts, and only the Security Administrator can do so. Accounts assigned to the Security Administrator role are used to manage Junos OS in accordance with [ND_cPP]. User accounts in the TOE have the following attributes: user identity (user name), authentication data (password) and role (privilege). The Security Administrator is associated with the defined login class "security-admin", which has the necessary permission set to permit the administrator to perform all tasks necessary to manage Junos OS in accordance with the requirements of [ND_cPP]. Accounts assigned to the Security Administrator role can also be used to access the Host OS. Direct access to these environments is disabled in the evaluated deployment, as a user has to first authenticate as a Security Administrator and can then access the underlying Wind River Linux Host OS . No direct console access to the hypervisor is permitted.
FPT_APW_EXT.1	Locally stored authentication credentials are protected: • The passwords are stored in obfuscated form using sha-1, sha-256 or sha-512. • 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.
FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3	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.

Requirement	TSS Description
	• File integrity test –verifies integrity of all mounted signed packages, to assert that system files have not been tampered with. To test the integrity of the firmware, the fingerprints of the executables and other immutable files are regenerated and validated against the SHA1 fingerprints contains in the manifest file. • Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys and iked credentials, such as Cas, CERTS, and various keys. • Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real. • Kernel, libmd, OpenSSL, QuickSec, SSH IPsec – verifies correct output from known answer tests for appropriate algorithms. Within the package, each Junos OS firmware image includes fingerprints of executables and other immutable files. Junos firmware will not execute any binary without validating a registered fingerprint. This feature protects the system against unauthorized software and activity that might compromise the integrity of the device. These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE. In the event of a transiently corrupt state or failure condition within the TOE, 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 applicable self-tests. When any self-test fails, the device halts in an error state. No command line input or traffic to any interface is processed. The device must be power cycled to attempt to return to operation.
FPT_SKP_EXT.1	Storage of pre-shared keys, symmetric keys, and private keys is described in Table 19. Junos OS does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission.
FPT_STM_EXT.1	All events recorded by syslog are timestamped. The clock function of Junos OS provides a source of date and time information for the appliance, used in audit timestamps. The clock is also used to determine certificate expiration, administrator session timeouts, and IPsec/SSH rekey thresholds. Wind River Linux kernel provides the current time when it bootstraps the Junos OS VM. Once the Junos OS VM is started it maintains its own time using the hardware Time Stamp Counter as the clock source.
FPT_TUD_EXT.1	Security Administrators are able to query the current version of the TOE firmware using the CLI command “show version” and, if a new version of the TOE firmware is available, initiate an update of the TOE firmware. Junos OS does not provide partial updates for the TOE, customers requiring updates must migrate to a subsequent release. Updates are downloaded and applied manually (there is no automatic updating of the Junos OS). The installable software package includes both the JCP VM (comprised of the Hypervisor and the Junos OS firmware) and the underlying Wind River Linux host OS. These cannot be updated separately in the evaluated configuration; they must be installed as a single package. Once the package is loaded the

Requirement	TSS Description
	procedures detailed in the Guidance document will be applied to disable the loading of additional VMs. The installable software package has a digital signature that is checked when the Security Administrator attempts to install the package. The firmware is digitally signed. The signature of the complete package is verified at the beginning of the installation process before the package is expanded. If signature verification fails, an error message is displayed and the package is not installed. The host device will reboot to completion the installation. The installation of the Wind River Linux OS is performed first. If the Wind River Linux kernel installation fails for any reason error log message will be output to the screen, and the system will halt waiting for administrator intervention (no audit event will be recorded at this time as the Junos environment is not running and the administrator will be aware of the failure due to the system halt) . Following successful installation of the Wind River Linux kernel the Junos VM installation will proceed. The Junos OS kernel maintains a set of fingerprints (SHA1 digests) for executable files and other files which should be immutable, as described in Section 7.3. The manifest file is signed using the Juniper package signing key and is verified by the TOE using the public key (stored on the TOE filesystem in clear, protected by filesystem access rights). ECDSA (P-256) with SHA-256 is used for digital signature package verification. The fingerprint loader will only process a manifest for which it can verify the signature. Thus, without a valid digital signature an executable cannot be run. When the command is issued to install an update, 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 uses the last known verified image.
FTA_SSL.3, FTA_SSL_EXT.1	The Security Administrator can set the TOE so that a user session is terminated after a period of inactivity. For each user session Junos OS 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. Junos OS 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 Security Administrator can configure this inactivity timer on administrative sessions after which the session will be logged out.
FTA_SSL.4	User sessions (local and remote) can be terminated by users. The administrative user can logout of existing session by typing logout to exit the CLI admin session and the Junos OS makes the current contents unreadable after the admin initiates the termination. No user activity can take place until the user re-identifies and authenticates.
FTA_TAB.1	Junos enables Security Administrators to configure an access banner provided with the authentication prompt. The banner can provide warnings against unauthorized access to the secure switch as well as any other information that

Requirement	TSS Description
	the Security Administrator wishes to communicate. The banner is shown at the local console and during remote access sessions using SSH.
FTP_ITC.1	Junos OS provides an SSH server to support Trusted Channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The remote audit server initiates the connection. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.
FTP_ITC.1/VPN	The TOE identifies as a multi-site VPN gateway. It has a dedicated IPsec VPN interface that only supports tunnel mode. NFX350 supports numerous routing standards as well as IPSec protocols. These functions can all be managed through the Junos OS software, either from a connected console on the management interface or via a network connection. Network management can be secured using IPsec (SSH is covered in FTP_ITC.1). Secure communication mechanism includes inbound and outbound traffic. For inbound traffic, the TOE looks up the SA by using the destination IP address, security protocol, and security parameter index (SPI) value. For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel. For allowed protocols the TOE supports AES-GCM-128, AES-GCM-192 and AES-GCM-256, and AES-CBC-128, AES-CBC-192 or AES-CBC-256 using HMAC SHA-256 for ESP protection. 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 no support for NAT traversal) and RFC 4868 for hash functions. IKEv1 aggressive mode is not supported. The TOE supports AES-CBC-128, AES-CBC-192, and AES-CBC-256 for payload protection in IKEv1 and IKEv2. The TOE also supports AES-GCM-128 and AES-GCM-256 for the payload protection in IKEv2.
FTP_TRP.1/Admin	The Junos OS SSH Server supports Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between Junos OS SSH Server and a remote administrator is provided by the use of an SSH session. Remote administrators of Junos OS initiate communication to the Junos CLI through the SSH tunnel created by the SSH session. Assured identification of Junos OS is guaranteed by using public key based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.
IPS_ABD_EXT.1, IPS_IPB_EXT.1, IPS_NTA_EXT.1, IPS_SBD_EXT.1	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

Requirement	TSS Description
	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 . 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 rule base. 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. 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. • 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: o No Action o Drop packet o Drop connection o Close client (send an RST packet to the client) o Close server (sends an RST packet to the server) o 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

Requirement	TSS Description
	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. The TOE is capable of inspecting IPv4, IPv6, ICMPv4, ICMPv6, TCP and UDP traffic. Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team. The TOE is capable of inspecting all traffic passing through the TOE's Ethernet interfaces (inline mode). Each of these interfaces types can be assigned to Zones on which firewall and IDP policies are predicated. The TOE supports the definition of known-good and known-bad lists of source and/or destination addresses at the firewall rule level. Address ranges can be defined by creating address book entries and attaching them to firewall policies. IPS signatures (in the sense of the MOD_IPS_V1.0) 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. • ICMPv4: 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". 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

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>
define custom-attack signatures for these application layer protocols using the command “set security idp custom-attack”.

The TOE is capable of detecting the following signatures using Junos predefined screen options:
<table>
<tr><th>MOD_IPS signature name</th><th>Junos screen name</th></tr>
<tr><td>IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)</td><td>ip tear-drop</td></tr>
<tr><td>IP source address equal to the IP destination (Land attack)</td><td>tcp land</td></tr>
<tr><td>Fragmented ICMP Traffic (e.g. Nuke attack)</td><td>icmp fragment</td></tr>
<tr><td>Large ICMP Traffic (Ping of Death attack)</td><td>icmp ping-death</td></tr>
<tr><td>TCP NULL flags</td><td>tcp tcp-no-flag</td></tr>
<tr><td>TCP SYN+FIN flags</td><td>tcp syn-fin</td></tr>
<tr><td>TCP FIN only flags</td><td>tcp fin-no-ack</td></tr>
<tr><td>UDP Bomb Attack</td><td>udp length-error</td></tr>
<tr><td>ICMP flooding (Smurf attack, and ping flood)</td><td>icmp flood</td></tr>
<tr><td>TCP flooding (e.g. SYN flood)</td><td>tcp syn-flood</td></tr>
<tr><td>IP protocol scanning</td><td>ip unknown-protocol</td></tr>
<tr><td>TCP port scanning</td><td>tcp port-scan</td></tr>
<tr><td>UDP port scanning</td><td>udp port-scan</td></tr>
<tr><td>ICMP scanning</td><td>icmp ip-sweep</td></tr>
</table>
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” ;
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), time of the day for defined source/destination address and source/destination port, frequency of traffic patterns and thresholds of traffic patterns.

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).
</td></tr>
</table>

Requirement	TSS Description
	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.

6.1 CAVP Algorithm Certificate Details

All FIPS-approved cryptographic functions implemented by the secure network appliance are implemented in the following two main libraries: Quicksec (for IKE), and OpenSSL libcrypto (for IPSec and, for SSH and the PKI daemon). The Junos OS Kernel (veriexec) library provides file-signing and verification for all executables on the TOE. All random number generation by the TOE is performed in accordance with NIST Special Publication 800-90 using HMAC_DRBG implemented in the OpenSSL library (FCS_RBG_EXT.1.1). Additionally, SHA (256, 512) is implemented in the LibMD library which is used for password hashing by Junos' MGD daemon. The network device is to be operated with FIPS mode enabled. The FIPS approved algorithms are applied when the FIPS operating mode is enabled.

Each of these cryptographic algorithms have been validated as identified in the table below.

Table 18 – CAVP Algorithm Certificate References

Crypto Module/ Library	FIPS PUB	Algorithm, Mode, Keysize, Function, Hashing, Usage	Certificate Number	Processor
OpenSSL libcrypto – OpenSSL IPSec Daemon	FIPS 197, SP 800-38D	AES-GCM (128, 192, 256) (Encrypt, Decrypt, AEAD)	#A2577	Intel® Xeon D-2146NT (Skylake)
	FIPS 197, SP 800-38A	AES-CBC (128, 192, 256) (Encrypt, Decrypt)	#A2577	
	FIPS 180-4	SHS: SHA (256) Byte Oriented (Message Digest Generation)	#A2577	
	FIPS 198-1	HMAC-SHA (256) Byte Oriented (Message Authentication)	#A2577	
Junos OS quicksec - ipsec-7 – IKED Daemon	FIPS 197, SP 800-38A	AES-CBC (128, 192, 256) (Encrypt, Decrypt)	A2576	
	FIPS 197, SP 800-38D	AES-GCM (128, 256) (Encrypt, Decrypt, AEAD)	A2576	

Crypto Module/ Library	FIPS PUB	Algorithm, Mode, Keysize, Function, Hashing, Usage	Certificate Number	Processor
	FIPS 180-4	SHS: SHA (256, 384) Byte Oriented (Message Digest Generation)	A2576	
	FIPS 198-1	HMAC-SHA (256, 384)	A2576	
	FIPS 186-4	RSA PKCS1_V1_5 (n=2048, 4096 (SHA 256) (SigGen, SigVer)	A2576	
	FIPS 186-4	ECDSA (P-256 w/ SHA-256) ECDSA (P-384 w/ SHA-384) (KeyGen, SigGen, SigVer)	A2576	
Junos OS quicksec - ipsec-7 – All Daemons	SP 800-90A	DRBG (HMAC-SHA-2-256) (Random Bit Generation)	A2576	
OpenSSL libcrypto – IKED Daemon	SP 800-56A	KAS ECC SSC (P-256, P-384, P-521)	#A2577	
OpenSSL libcrypto – SSHD Daemon and PKID Daemon	FIPS 197, SP800-38A	AES-CBC/CTR (128, 256) (Encrypt, Decrypt)	#A2577	
	FIPS 180-4	SHS: SHA (1, 256, 384) Byte Oriented (Message Digest Generation, SSH KDF Primitive)	#A2577	
	FIPS 180-4	SHS: SHA-512 Byte Oriented (Message Digest Generation)	#A2577	
	FIPS 198-1	HMAC-SHA (1, 256), (512) Byte Oriented (Message Authentication DRBG Primitive)	#A2577	
	SP 800-56A	KAS ECC SSC (P-256, P-384, P-521)	#A2577	
OpenSSL libcrypto – All Daemons	SP 800-90A	DRBG (HMAC-SHA-2-256) (Random Bit Generation)	#A2577	
	FIPS 186-4	RSA KeyGen (n=2048 (SHA-256)	#A2577	

Crypto Module/ Library	FIPS PUB	Algorithm, Mode, Keysize, Function, Hashing, Usage	Certificate Number	Processor
	FIPS 186-4	RSA PKCS1_V1_5 (n=2048, 4096 (SHA-256) (SigGen, SigVer)	#A2577	
	FIPS 186-4	ECDSA [P-256 (SHA-256)], [P-384 (SHA-384)], [P-521 (SHA-521)]] (SigGen, SigVer, KeyGen, KeyVer)	#A2577	
Junos OS libmd – MGD Daemon, Password Hashing	FIPS 198-1	HMAC-SHA (1, 256) Byte Oriented (Message Authentication DRBG Primitive)	A2575	
	FIPS 180-4	SHS: SHA (256, 512) Byte Oriented (Message Digest Generation)	A2575	
Junos OS Kernel – Veriexec	FIPS 180-4	SHS: SHA (1, 256) Byte Oriented (Message Digest Generation)	A2574	
Junos OS Kernel – kernel-hmac drbg	FIPS 198-1	HMAC-SHA (256) Byte Oriented (DRBG Primitive)	A2574	
Junos OS Kernel – kernel-hmac drbg	SP800-90A	DRBG (HMAC-2-SHA-256) (Random Bit Generation)	A2574	

6.2 Cryptographic Key Descriptions

The table below describes the keys provided by the TOE.

Table 19 – Key Descriptions

Keys/CSPs	Purpose	Method of Storage	Storage Location	Method of Zeroization
SSH Private Host Key	The first time SSH is configured the set of Host keys is generated. Used to identify the host. ecdsa-sha2-nistp256 (ECDSA P-256, ECDSA P-384, ECDSA P-521)	Plaintext	File format on Disk (mapped to SDD)	When the appliance is recommissioned, the config files (including CSP files such as SSH keys) are removed using the Linux `shred` command to wipe the underlying

Keys/CSPs	Purpose	Method of Storage	Storage Location	Method of Zeroization
	and/or ssh-rsa (RSA 2048)			persistent storage media.
SSH Private Host Key	Loaded into memory to complete session establishment	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination (when released by the Junos VM, the WRL hypervisor erases the released memory before it is placed in the free pool)
SSH Session Key	Session keys used with SSH, AES 128, 256, hmac-sha-1, hmac-sha2-256 or hmac-sha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination (when released by the Junos VM, the hypervisor releases the memory and places it in the free pool)
User Password	Plaintext value as entered by user	Plaintext as entered	Processed in Memory	Memory free() operation is performed by Junos (when released by the Junos VM, the hypervisor releases the memory and places it in the free pool)
		Hashed when stored (HMAC-sha1, sha256, sha512)	Stored on disk (mapped to SDD)	When the appliance is recommissioned, the config files (including the obfuscated password) are removed using the “request system zeroize” option.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero’s at reboot.
IKE Private Host key	Private authentication key	Plaintext	Disk (mapped to SDD)/Memory	‘clear security IKE security-association’

Keys/CSPs	Purpose	Method of Storage	Storage Location	Method of Zeroization
	used in IKE. RSA 2048, RSA 4096, ECDSA P-256, ECDSA P-384			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-association’ or reboot the box.
IKE-DH Private Exponent	Ephemeral DH private exponent used in IKE. DH N = 224 bit or N = 256 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 (mapped to SDD)/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.
ecdh private keys	Loaded into memory to complete key exchange in session establishment	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination (when released by the Junos VM, the WRL hypervisor erases the released memory before it is placed in the free pool)

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 20 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CC	Common Criteria
CRL	Certificate Revocation List
DTLS	Datagram Transport Layer Security
EP	Extended Package
GUI	Graphical User Interface
IP	Internet Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PP	Protection Profile
RSA	Rivest, Shamir, & Adleman
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
CLI	Command Line Interface
VM	Virtual Machine
VNF	Virtualized Network Functions
VPN	Virtual Private Network