Juniper NFX series Network Services Platform with Junos OS 23.4R1 Security Target

Document Version: 1.0

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History:

Version	Date	Changes
Version 0.1	18 January 2024	Initial draft
Version 0.2	02 February 2024	Updated as per internal review comments.
Version 0.3	12 March 2024	Updated as per inputs from vendor.
Version 0.4	09 October 2024	Testing-based modifications and TD updates
Version 0.5	28 October 2024	Updated to address internal review comments
Version 0.6	01 November 2024	Minor update to the PP config section
Version 0.7	30 December 2024	Updated to address check-in ECR comments
Version 0.8	23 April 2025	Minor updates to the CAVP and TSS sections, removal of PSK claims for IPsec and application of latest TDs.
Version 0.9	26 May 2025	Updates to address gaps identified by the AAR.
Version 1.0	07 October 2025	Minor updates to address 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.

Category	Identifier
ST Title	Juniper NFX series Network Services Platform with Junos OS 23.4R1 Security Target
ST Version	1.0
ST Date	07 October 2025
ST Author	Acumen Security, LLC.
TOE Identifier	Juniper NFX series Network Services Platform with Junos OS 23.4R1
TOE Version	23.4R1
TOE Developer	Juniper Networks, Inc.
Key Words	Network Device, VPN Gateways, IPS, Firewall

Table 1 TOE/ST Identification

1.2 TOE Overview

The TOE is Juniper Networks, Inc. NFX series Network Services Platform with Junos OS 23.4R1. The NFX series devices integrate routing, switching, and security functions on a single platform.

The devices support the definition of, and enforce, 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 flow of information 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, firewall functionality and also implements Intrusion Prevention System (IPS) 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 NFX series with Junos OS 23.4R1 TOE includes a hypervisor, which runs a virtual machine (VM) on an NFX series hardware model:

- NFX150
    - NFX150-C-S1

  - NFX150-S1
  - NFX150-S1E
- NFX250
  - NFX250-S1
  - NFX250-S1E
  - NFX250-S2
- NFX350
  - NFX350-S1
  - NFX350-S2
  - NFX350-S3

1.3 TOE Description

1.3.1 Physical boundary

The physical boundaries of the TOE is the NFX series hardware running Junos OS 23.4R1. The Junos OS 23.4R1 software includes the KVM Hypervisor as well as the JCP and JDM application container. Hence the TOE is contained within the physical boundary of each platform specified in Section 1.3.3. The TOE is delivered as a single device with the Junos OS software installed. There are several mechanisms provided in the delivery process to ensure that a customer receives a product that has not been tampered with, such as use of shipping labels, well-sealed packaging and email notification containing details such as the carrier tracking number, that can be compared with the ones on the received shipment. 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.
Figure 1 below shows the general architecture for the NFX150.

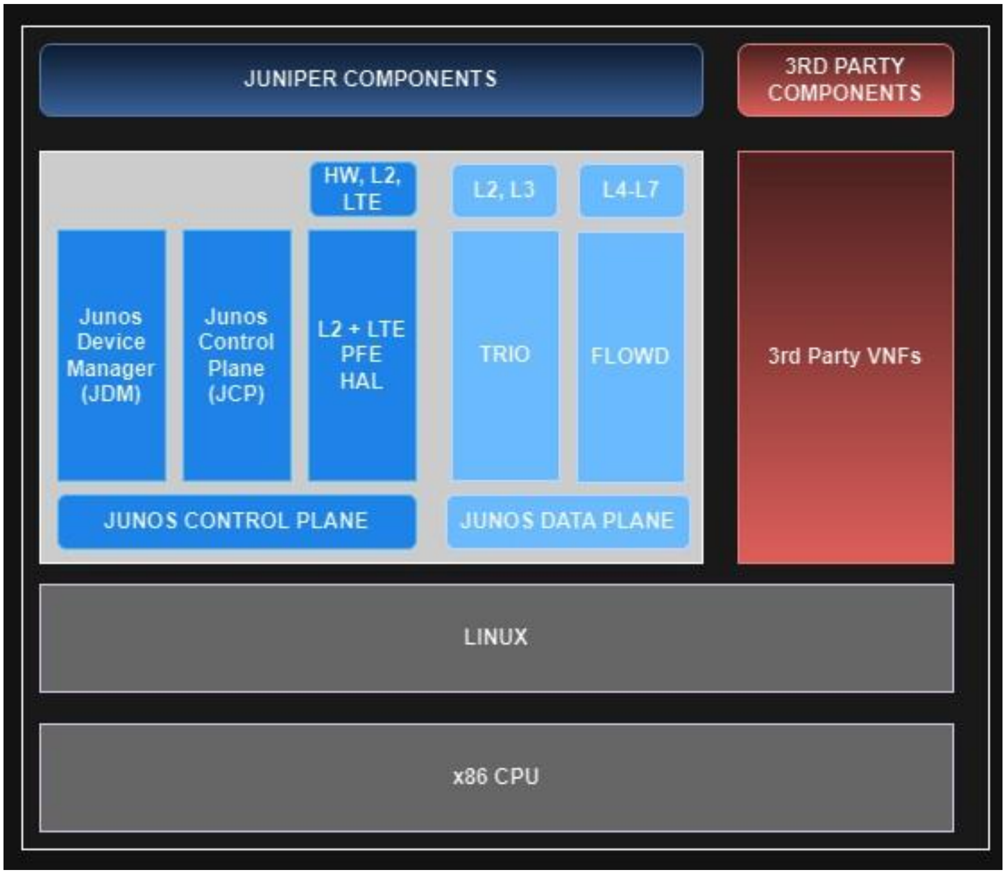

Figure 1 NFX150 Software Architecture

Figure 2 below shows the general architecture for the NFX250 and NFX350.

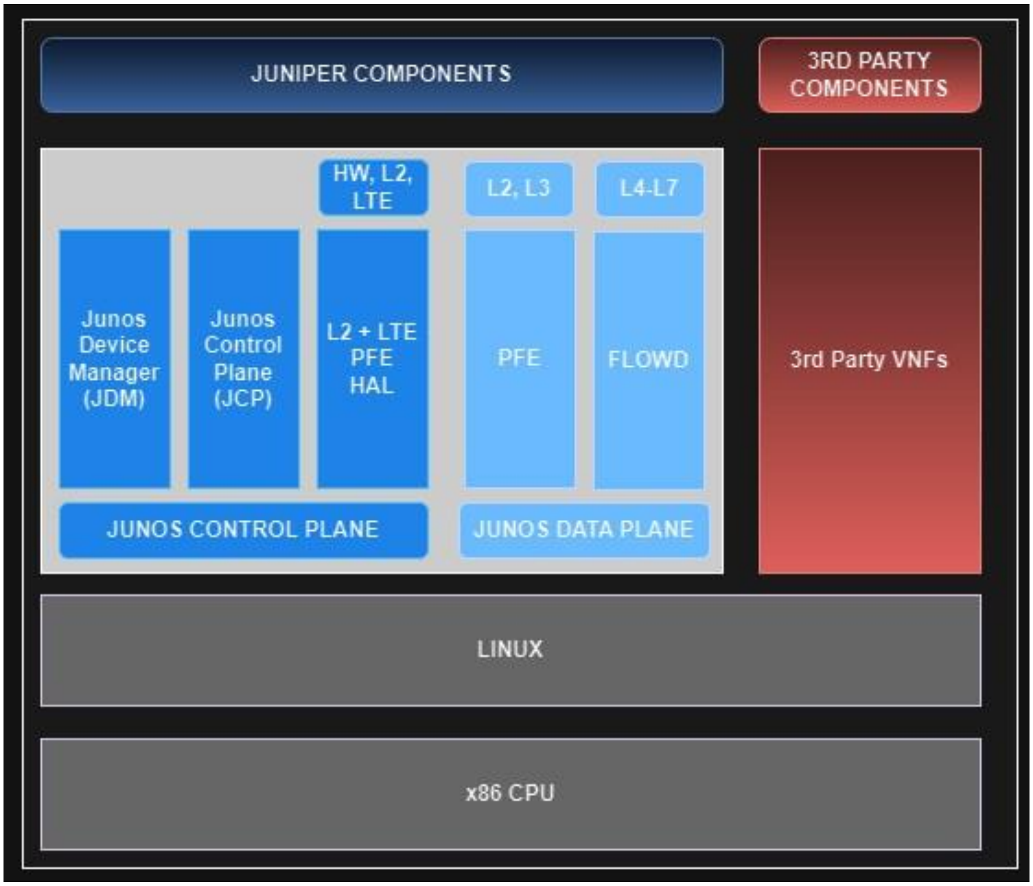

Figure 2 NFX250/NFX350 Software Architecture

All the NFX devices support the installation of 3rd party VMs, but installation of 3rd party VMs is not allowed in the evaluated configuration.

1.3.1.1 Juniper Device Manager

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

1.3.1.2 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 NFX devices, and JCP runs by default as vjunos0. 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 NFX 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.

1.3.1.2.1 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.

1.3.1.2.2 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.

1.3.1.3 JCP Administration

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

1.3.1.4 Linux

The NFX devices run on Wind River Linux LTS 19 as their host OS. The host OS functions as a hypervisor and runs natively on an Intel x86 processor.

1.3.2 Logical boundary

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

Functionality	Description
Protected Communications	The TOE provides an SSH server to support protected communications for administrators to establish secure management sessions and to connect to external syslog servers. The TOE also supports IPsec connections to provide multi-site virtual private network (VPN) gateway functionality. The TOE requires that

Functionality	Description
	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 inactive 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 using digital signatures.
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 14, Table 15 and Table 16. 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 are 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 NFX services. Access to manage the device’s FreeBSD host can only be gained through the JCP.

Functionality	Description
Packet Filtering/Stateful Traffic Filtering	The TOE provides stateful network traffic filtering for IP-based (IPv4 as well as IPv6) traffic, based on examination of network packets and the application of information flow rules. This functionality is common across all three NFX devices and involves no cryptographic operations in terms of processing.
Intrusion Prevention	The TOE can be configured to analyze IP-based (IPv4 as well as IPv6) 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. The IPS functionality is common across all three NFX devices and involves no cryptographic operations in terms of processing.
User Data Protection/Information Flow Control	The TOE is designed to forward IP-based (IPv4 as well as IPv6) 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).

Table 2 TOE Logical Boundary

1.3.3 Product Functionality not Included in the Scope of the Evaluation

The following product functionalities are disabled in the CC-evaluated configuration of the TOE:

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

The following product functionalities are supported by the product but not covered by the CC evaluation and are hence expected to not be used when deployed in a CC configuration:

- Use of SNMP, since it violates the Trusted Path requirement set
- Use of the Junos or Linux root account.
- Hosting additional VMs on the TOE physical platform.
- Use of routing protocols such as OSPF, BGP and RIP.

In general, any product functionality not covered by the Protection Profile, Modules or Packages mentioned in Section 2.2 are not covered by the CC evaluation.

1.3.4 Hardware

The hardware model specifications for the 3 devices are described in the tables below.

1.3.4.1 NFX150

Specification	NFX150-C-S1	NFX150-S1	NFX150-S1-E
Form Factor	Desktop	Rack-mount	
Rack units (U)	1 U	1 U	
Power	75 W AC-DC Power Adapter	150W AC-DC open frame power	
CPU	Intel 4 Core ATOM C3500 series	Intel 8 Core ATOM C3700 series	
Micro-Architecture	Denverton	Denverton	
Memory	8 GB DDR4	16 GB DDR4	32 GB DDR4
Storage	100 GB SSD	200 GB SSD	
Host OS	Wind River Linux LTS 19	Wind River Linux LTS 19	
Integrated network interfaces	4 x 10/100/1000BASE-T RJ-45 LAN ports 2 x 1GbE/10GbE SFP+ WAN ports 1 x 10/100/1000BASE-T RJ-45 management port		
Managed Secure Router	200 Mbps	500 Mbps	800 Mbps
Managed Security	200 Mbps	500 Mbps	800 Mbps
IPsec	80 Mbps	150 Mbps	300 Mbps
Out-of-band interfaces	RJ-45 console port Mini USB console port USB 3.0 port		
Junos OS Kernel	FreeBSD 12.1		
Max VNFs	1 - 2	2 – 3	

Table 3 NFX150 Hardware

1.3.4.2 NFX250

Specification	NFX250-S1	NFX250-S1E	NFX250-S2
Rack units (U)	1 U	1 U	1 U
Power	Fixed PSU 100-240 VAC	Fixed PSU 100-240 VAC	Fixed PSU 100-240 VAC
CPU	Intel 6 Core Xeon D-1528	Intel 6 Core Xeon D-1528	Intel 6 Core Xeon D-1528
Micro-Architecture	Broadwell	Broadwell	Broadwell
Memory	16 GB DDR4	16 GB DDR4	32 GB DDR4
Storage	100 GB[1] SSD	200 GB[1] SSD	400 GB[1] SSD
Host OS	Wind River Linux LTS 19	Wind River Linux LTS 19	Wind River Linux LTS 19

[1] Raw capacity; actual capacity will be lower due to overprovisioning.

Specification	NFX250-S1	NFX250-S1E	NFX250-S2
Network interfaces	• 8 x 10/100/1000BASE-T RJ-45 LAN ports • 2 x 10/100/1000BASE-T RJ-45 LAN/WAN ports • 2 x 100/1000BASE-X small form-factor pluggable transceiver (SFP) WAN ports • 2 x 1GbE/10GbE SFP+ WAN ports • 1 x 10/100/1000BASE-T RJ-45 management port • ADSL2/VDSL2 SFP[2]	• 8 x 10/100/1000BASE-T RJ-45 LAN ports • 2 x 10/100/1000BASE-T RJ-45 LAN/WAN ports • 2 x 100/1000BASE-X small form-factor pluggable transceiver (SFP) WAN ports • 2 x 1GbE/10GbE SFP+ WAN ports • 1 x 10/100/1000BASE-T RJ-45 management port • ADSL2/VDSL2 SFP[2]	• 8 x 10/100/1000BASE-T RJ-45 LAN ports • 2 x 10/100/1000BASE-T RJ-45 LAN/WAN ports • 2 x 100/1000BASE-X SFP WAN ports • 2 x 1GbE/10GbE SFP+ WAN ports • 1 x 10/100/1000BASE-T RJ-45 management port • ADSL2/VDSL2 SFP[2]
Managed Secure Router	2 Gbps	3 Gbps	4 Gbps
Managed Security	2 Gbps	3 Gbps	4 Gbps
IPsec	500 Mbps	750 Mbps	1.2 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
Junos OS Kernel	FreeBSD 12.1		
Maximum number of VNFs	6	6	8

Table 4 NFX250 Hardware

1.3.4.3 NFX350

Specification	NFX350-S1	NFX350-S2	NFX350-S3
Rack units (U)	1 U	1 U	1 U
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

[2] ADSL2/VDSL2 interfaces are provided by a small form-factor pluggable transceiver which can be used in any SFP port on the NFX250.

Specification	NFX350-S1	NFX350-S2	NFX350-S3
Memory	32 GB DDR4	64 GB DDR4	128 GB DDR4
Storage	100 GB SSD	100 GB SSD	100 GB SSD
Host OS	Wind River Linux LTS 19	Wind River Linux LTS 19	Wind River Linux LTS 19
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 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 1GbE/10GbE • SFP+ LAN or WAN ports • 1 x 10/100/ • 1000BASE-T RJ-45 management port
Managed Secure Router	12 Gbps	20 Gbps	30 Gbps
Managed Security	12 Gbps	20 Gbps	30 Gbps
IPsec	2.5 Gbps	5 Gbps	7.5 Gbps
Out-of-band interfaces	RJ-45 console port Mini USB console port 2 x USB 3.0 port	RJ-45 console port Mini USB console port 2 x USB 3.0 port	RJ-45 console port Mini USB console port 2 x USB 3.0 port
Junos OS Kernel	FreeBSD 12.1		
Maximum number of VNFs	8	10	12

Table 5 NFX350 Hardware

1.3.5 TOE Documentation

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

- Junos OS Common Criteria Configuration Guide for NFX150, NFX250, and NFX350 Network Services Platforms (Release 23.4R1)
- Junos OS Intrusion Detection and Prevention User Guide (Published 2025-03-28)
- IPsec VPN User Guide (Published 2025-06-23)

1.4 TOE Environment

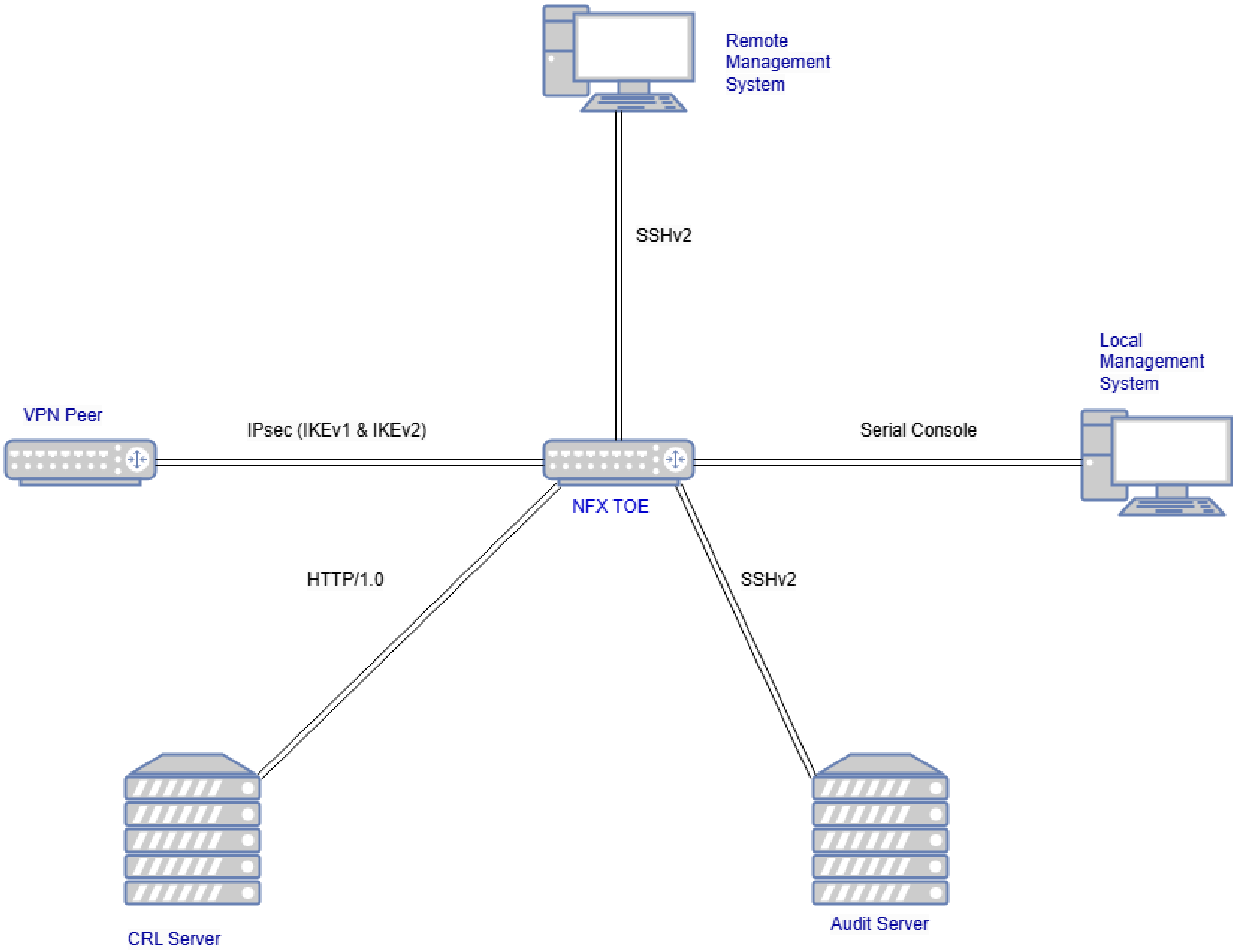

Figure 3 TOE Deployment Diagram

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

Components	Mandatory/Optional	Description
Remote Management System	Mandatory	The remote management system is used by an administrator to establish a connection using an SSHv2 client to configure the TOE.
Local Management System	Mandatory	The local management system is used by an administrator to configure the TOE over a serial console connection.

Components	Mandatory/Optional	Description
Audit Server	Mandatory	The audit server supports an SSHv2 client which inititates the trusted channel between itself and the TOE for transmission of logs using the netconf utility.
VPN Peer	Optional	The VPN peer is a dedicated IPsec interface that establishes an IPsec tunnel with the VPN gateway of the TOE, which supports both IKEv1 and IKEv2.
CRL Server	Mandatory	The CRL server, over HTTP/1.0, supports revocation checking of certificates used by the TOE for IPsec tunnels.

Table 6 Required Environmental Components

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 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 (Conformant)

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- PP-Configuration for Network Devices, Intrusion Protection Systems, Stateful Traffic Filter Firewalls, and Virtual Private Network Gateways, Version 2.0 (CFG_NDcPP-IPS-FW-VPNGW_V2.0). This PP-Configuration includes the following components:
    - Base PP: collaborative Protection Profile for Network Devices, version 3.0e, dated 06 December 2023 (CPP_ND_V3.0E),
    - PP-Module: collaborative Protection Profile Module for Stateful Traffic Filter Firewalls, Version 1.4e, dated 25 June 2020 (MOD_CPP_FW_V1.4E),
    - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways, version 1.3, dated 16 August 2023 (MOD_VPNGW_V1.3).
    - PP-Module: PP-Module for Intrusion Prevention Systems (IPS), Version 1.0, dated 11 May 2021 (MOD_IPS_V1.0).

- Package: Functional Package for Secure Shell (SSH), Version 1.0, dated 13 May 2021 (PKG_SSH_V1.0)

2.3 Conformance Rationale

This ST provides exact conformance to to the PP and Modules mentioned in Section 2.2. 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.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v3.0e, Firewall Module v1.4e, VPN Gateway Module v1.3, IPS Module v1.0 and SSH Functional Package v1.0 have been considered. Table 7 identifies all applicable TDs.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0545: NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	Yes	
TD0551: NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	Yes	
TD0595: Administrative corrections to IPS PP-Module	Yes	
TD0682: Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	
TD0695: Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
TD0722: IPS_SBD_EXT.1.1 EA Correction	Yes	
TD0732: FCS_SSHS_EXT.1.3 Test 2 Update	Yes	
TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
TD0781: Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3	No	Pre-shared keys (FIA_PSK_EXT.3) are not claimed.
TD0811: Correction to Referenced SFR in FIA_PSK_EXT.3 Test	No	Pre-shared keys (FIA_PSK_EXT.3) are not claimed.
TD0824: Aligning MOD_VPNGW 1.3 with NDcPP 3.0E	Yes	
TD0827: Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E	Yes	
TD0828: Aligning MOD_IPS_V1.0 with CPP_ND_V3.0E	Yes	
TD0836: NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Yes	
TD0838: PPK Configurability in FIA_PSK_EXT.1.1	No	Pre-shared keys (FIA_PSK_EXT.1) are not claimed.
TD0868: NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	Yes	
TD0879: NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Yes	
TD0880: NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Yes	
TD0886: Clarification to FAU_STG_EXT.1 Test 6	Y	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0899: NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	No	TLS functionality is not claimed.
TD0900: NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Yes	
TD0902: Updating RFC 2460 to 8200 in MOD_IPS_V1.0	Yes	
TD0909: Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Yes	
TD0921: NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Yes	
TD0923: NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Yes	TD is applicable but not relevant since the TOE does allow the administrator to configure local audit settings.
TD0924: NIT Technical Decision: FFW_RUL_EXT.1.2 Expected Rule Granularity Level	Yes	
TD0944: Adding FIPS 186-5 in MOD_VPNGW_V1.3	Yes	

Table 7 Relevant Technical Decisions

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 8 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

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

ID	Threat
	network traffic is exposed and there could be a loss of 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.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.
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.

ID	Threat
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 (VPNGW)	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 (VPNGW)	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

ID	Threat
	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 (VPNGW)	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 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.

ID	Threat
T.NETWORK_MISUSE (VPNGW)	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 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 (VPNGW)	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.

Table 8 Threats

3.2 Assumptions

The assumptions included in Table 9 are drawn directly from PP and any relevant EPs/Modules/Packages.

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.
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).

[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.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).
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 (VPNGW)	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

Table 9 Assumptions

3.3 Organizational Security Policies

The OSPs included in Table 10 are drawn directly from the PP and any relevant EPs/Modules/Packages.

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

Table 10 OSPs

4 Security Objectives

The security objectives have been taken directly from the claimed Base PP, PP-Modules, and Package, 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.

ID	Security Objectives
O.RESIDUAL_INFORMATION (MOD_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 (MOD_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 (MOD_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 (MOD_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 (MOD_IPS)	The IPS must respond appropriately to its analytical conclusions about IPS policy violations.
O.TOE_ADMINISTRATION (MOD_IPS)	The IPS will provide a method for authorized administrator to configure the TSF.
O.ADDRESS_FILTERING (MOD_VPNGW)	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 capability will restrict the flow of network traffic between protected networks and other attached networks based

ID	Security Objectives
	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 (MOD_VPNGW)	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 (MOD_VPNGW)	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 (MOD_VPNGW)	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 (MOD_VPNGW)	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 (MOD_VPNGW)	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 between peer VPN gateways, but also with certification authorities (CAs).

ID	Security Objectives
O.TOE_ADMINISTRATION (MOD_VPNGW)	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.

Table 11 Security Objectives for 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.

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 (MOD_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 (MOD_VPNGW)	The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

Table 12 Security Objectives for the Operational Environment

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.

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.1/IPS	Audit Data Generation (IPS)
FAU_GEN.1/VPN	Audit Data Generation (VPN)
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_SSH_EXT.1	SSH Protocol
FCS_SSHS_EXT.1	SSH Protocol - Server
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
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
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

Table 13 SFRs

5.1 Conventions

The conventions used in descriptions of the SFRs are as follows:

- Unaltered SFRs are stated in the form used in [CC2] or their extended component definition (ECD);
- Refinement made in the PP: the refinement text is indicated with **bold text** and ~~strikethroughs~~;
- Selection wholly or partially completed in the PP: the selection values (i.e. the selection values adopted in the PP or the remaining selection values available for the ST) are indicated with underlined text.

e.g. '[selection: disclosure, modification, loss of use]' in [CC2] or an ECD might become 'disclosure' (completion) or '[selection: disclosure, modification]' (partial completion) in the PP;

- Assignment wholly or partially completed in the PP: indicated with *italicized text*;
- Assignment completed within a selection in the PP: the completed assignment text is indicated with *italicized and underlined text*
  e.g. [selection: *change_default, query, modify, delete*, *[assignment: other operations]*]' in [CC2] or an ECD might become 'change_default, *select_tag*' (completion of both selection and assignment) or '[selection: change_default, *select_tag, select_value*]' (partial completion of selection, and completion of assignment) in the PP;
- Iteration: indicated by adding a string starting with '/' (e.g. 'FCS_COP.1/Hash').

Extended SFRs are identified by having a label 'EXT' at the end of the SFR 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) All 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 14***.

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).

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 14***.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
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
FCS_COP.1/SigGen	None	None
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_RBG_EXT.1	None	None
FDP_RIP.2	None	None
FCS_SSH_EXT.1	• Failure to establish an SSH session • Establishment of SSH connection • Termination of SSH connection session • Dropping of packet(s) outside defined size limits	• Reason for failure and Non-TOE endpoint of attempted connection (IP Address) • Non-TOE endpoint of connection (IP Address) • Non-TOE endpoint of connection (IP Address) • Packet size
FCS_SSHS_EXT.1	No events specified	N/A
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_UAU.7	None	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)

Requirement	Auditable Events	Additional Audit Record Contents
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	None	None
FMT_MTD.1/CryptoKeys	None	None
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
FPT_APW_EXT.1	None	None
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_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

Requirement	Auditable Events	Additional Audit Record Contents
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
FTP_ITC.1	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

Table 14 Security Functional Requirements and Auditable Events

Application note: In addition to the above table, FAU_GEN.1.1 defines further auditable events. TD0923 is applicable but not relevant since the TOE does allow the administrator to configure local audit settings.

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

FAU_GEN.1.1/IPS
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 dissimilar IPS events;*

d) *All dissimilar IPS reactions;*

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

f) *Totals of similar reactions occurring within a specified time period*.

g) *The events in the IPS Events table.*

h) *[no other auditable events]*]

Application note: From [MOD_IPS_V1.0]

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*].

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).
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).

Requirement	Auditable Events	Additional Audit Record Contents
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).

Table 15 MOD_IPS_v1.0 Security Functional Requirements and Auditable Events

Application note: From [MOD_IPS_V1.0]. Moreover, in addition to the above table, FAU_GEN.1.1/IPS defines further auditable events.

5.2.1.3 FAU_GEN.1/VPN Audit Data Generation (VPN Gateway)

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

a. Start-up and shutdown of the audit functions
b. Indication that TSF self-test was completed
c. Failure of self-test
d. All auditable events for the [*not specified*] level of audit; and
e. [*auditable events defined in the Auditable Events for Mandatory Requirements table*].

FAU_GEN.1.2/VPN

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 (if applicable), and the outcome (success or failure) of the event; and

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [additional information defined in the Auditable Events for Mandatory Requirements table for each auditable event, where applicable].

Application note: From [MOD_VPNGW_V1.3]

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/VPN	No events specified	N/A
FCS_CKM.1/IKE	No events specified	N/A
FMT_SMF.1/VPN	All administrative actions	No additional information.

Requirement	Auditable Events	Additional Audit Record Contents
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol
FPT_FLS.1/SelfTest	No events specified	N/A
FPT_TST_EXT.3	No events specified	N/A
FTP_ITC.1/VPN	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	• No additional information. • No additional information. • Identification of the initiator and target of failed trusted channel establishment attempt

Table 16 MOD_VPNGW_v1.3 Security Functional Requirements and Auditable Events

Application note: From [MOD_VPNGW_v1.3]. Moreover, in addition to the above table, FAU_GEN.1.1/VPN defines further auditable events.

5.2.1.4 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.

5.2.1.5 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 maintain a [log file] of audit records in the event that an interruption of communication with the remote audit server occurs

FAU_STG_EXT.1.4
The TSF shall be able to store [persistent] audit records locally with a minimum storage size of [*65536 bytes*].

FAU_STG_EXT.1.5
The TSF shall [overwrite previous audit records according to the following rule: [*oldest log entry is overwritten*]] when the local storage space for audit data is full.

FAU_STG_EXT.1.6
The TSF shall provide the following mechanisms for administrative access to locally stored audit records [manual export, ability to view locally].

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 bits, 3072 bits, or 4096-bits*] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- 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 , or FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2, or ISO/IEC 14888-3, “IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms”, Section 6.6;
- 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].

]

Application note: TD0921 has been applied.

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 186-5, "Digital Signature Standard (DSS)", Appendix A.1 for RSA schemes;**
- **FIPS PUB 186-5, “Digital Signature Standard (DSS),” Appendix A.2 for ECDSA schemes, and implementing “NIST curves” P-384 and [P-256]**

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].

Application note: From [MOD_VPNGW_V1.3] with TD0944 applied.

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 [groups listed in 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].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

5.2.2.6 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*.

5.2.2.7 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-512, implicit] and cryptographic key sizes *[160, 256, and 512 bits]* **and message digest sizes** [160, 256, 512] **bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 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,
- *Elliptic Curve Digital Signature Algorithm*

]
and cryptographic key sizes [

- For RSA: [modulus 2048, 3072 or 4096 bits],
- For ECDSA: [256, 384 or 521 bits]

]
that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 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 implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.
  ].

Application note: TD0921 has been applied.

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, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)] and [AES-CBC-192 (specified in RFC 3602)] together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-1, HMAC-SHA-256].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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 numbers] and [RFC 4868 for hash functions];
- IKEv2 as defined in RFC 7296 [with no support for NAT traversal], and [RFC 4868 for hash functions]

].

Application note: From [MOD_VPNGW_V1.3] and has been updated as per TD0824.

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)].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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 between [*0.05 hours*] and [*24 hours*];

];

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

  ]

].

Application note: TD0868 has been applied.

FCS_IPSEC_EXT.1.8
The TSF shall ensure that [

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

    ]

].

Application note: TD0868 has been applied.

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 *[224 (for DH Group 14), 256 (for DH Group 19) or 384 (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;

].

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

].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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 [**IKEv1, IKEv2**] protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

Application note: From [MOD_VPNGW_V1.3] and has been updated as per TD0824.

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]].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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 [SHA-256]].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[1]* software-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_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1
The TOE shall implement SSH acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8332] and [*no other standard*].

Application note: Updated as per TD0909.

FCS_SSH_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods: [

- “password” (RFC 4242),
- “keyboard-interactive” (RFC 4256),
- “publickey” (RFC 4252): [
    - ssh-rsa (RFC 4253),
    - rsa-sha2-256 (RFC 8332),
    - rsa-sha2-512 (RFC 8332),
    - ecdsa-sha2-nistp256 (RFC 5656),
    - ecdsa-sha2-nistp384 (RFC 5656),
    - ecdsa-sha2-nistp521 (RFC 5656)

    ]

] and no other methods.

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

FCS_SSH_EXT.1.4
The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- aes128-ctr (RFC 4344),
- aes256-ctr (RFC 4344),
- aes128-cbc (RFC 4253),
- aes256-cbc (RFC 4253)

] and no other mechanisms.

FCS_SSH_EXT.1.5
The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668),
- hmac-sha2-512 (RFC 6668)

] and no other mechanisms.

FCS_SSH_EXT.1.6
The TSF shall establish a shared secret with its peer using: [

- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),
- ecdh-sha2-nistp521 (RFC 5656)

] and no other mechanisms.

FCS_SSH_EXT.1.7
The TSF shall use SSH KDF as defined in [

- RFC 5656 (Section 4)

] to derive the following cryptographic keys from a shared secret: *session keys*.

FCS_SSH_EXT.1.8
The TSF shall ensure that [

- a rekey of the session keys

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

5.2.2.12 FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1
The TSF shall authenticate itself to its peer (SSH Client) using: [

- ssh-rsa (RFC 4253),
- rsa-sha2-256 (RFC 8332),
- rsa-sha2-512 (RFC 8332),
- ecdsa-sha2-nistp256 (RFC 5656),
- ecdsa-sha2-nistp384 (RFC 5656),
- ecdsa-sha2-nistp521 (RFC 5656)

].

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 *[2 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 *[the action of executing the unlocking command]* is taken by an Administrator; 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: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [“"”, ”'”, ”+”, “,”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “?”, “[”, “\”, “]”, “_”, “`”, “{”, “|”, “}”, “~”]];

b) Minimum password length shall be configurable to between *[6]* 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 reply*].

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.

FIA_UIA_EXT.1.3

The TSF shall provide the following remote authentication mechanisms [SSH password, SSH public key] and [no other mechanism]. The TSF shall provide the following local authentication mechanisms [password-based].

Application note: TD0900 has been applied.

FIA_UIA_EXT.1.4

The TSF shall authenticate any administrative user’s claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

5.2.3.4 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**.

5.2.3.5 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.

5.2.3.6 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 other protocols]**, and [no additional uses].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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].

5.2.3.7 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.

5.2.4 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 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*].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

5.2.4.6 FMT_SMF.1 Specification of Management Functions

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

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;

- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [
    - Ability to start and stop services;
    - Ability to configure local audit behavior (e.g. changes to storage locations for audit; changes to behavior when local audit storage space is full;changes to local audit storage size);
    - Ability to modify the behavior of the transmission of audit data to an external IT entity;
    - Ability to manage the cryptographic keys;
    - Ability to manage the cryptographic functionality;
    - Ability to configure thresholds for SSH rekeying;
    - Ability to configure the lifetime for IPsec SAs;
    - 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 administer the TOE locally;
    - Ability to configure the local session inactivity time before session termination or locking;
    - Ability to configure the authentication failure parameters for FIA_AFL.1;
    - Ability to manage the trusted public keys database;

    ]

Application note: TD0880 has been applied.

5.2.4.7 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*]

Application note: From [MOD_IPS_V1.0]

5.2.4.8 FMT_SMF.1/VPN Specification of Management Functions (VPN Gateway)

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

- Definition of packet filtering rules
- Association of packet filtering rules to network interfaces
- Ordering of packet filtering rules by priority

[

- No other capabilities

]].

Application note: From [MOD_VPNGW_V1.3]

5.2.4.9 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;*

Application note: From [MOD_CPP_FW_V1.4E]

5.2.4.10 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 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 8200)
    - 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 FPT_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 start-up (on power on) to verify the integrity of the TOE firmware and software;

• Prior to providing any cryptographic service and [on-demand] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;

• [start-up, continuous] self-tests to demonstrate the correct operation of the TSF: noise source health tests.

Application note: From [MOD_VPNGW_V1.3] and as per TD0824, superseding the verbiage as per TD0836.

FPT_TST_EXT.1.2
The TSF shall respond to [all failures] by [rebooting].

Application note: From [MOD_VPNGW_V1.3] and as per TD0824.

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*].

Application note: From [MOD_VPNGW_V1.3].

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

Application note: From [MOD_VPNGW_V1.3] and has been updated as per TD0824.

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*].

Application note: From [MOD_VPNGW_V1.3].

5.2.7 TOE Access (FTA)

Application note: TD0879 has been applied.

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 capabiltites]** 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 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].

Application note: From [MOD_VPNGW_V1.3].

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.

Application note: From [CPP_FW_V1.4e].

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;

]

and the following network protocol fields:

- *[[IPv4: source address; destination address*
- *IPv6: source address; destination address*
- *TCP: source port; destination port*
- *UDP: source port; destination port]]*

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

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

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

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 8200
- 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].

Application note: TD0902 has been applied.

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: [*Gigabit Ethernet interfaces*];
- Management mode: [*FastEthernet interface: dedicated management Ethernet interface*];
    - [Session-reset-capable interfaces: [*Gigabit 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 [*ID, sequence number*].
- 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;]
- 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 below Table 17.

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

Assurance Class	Assurance Components	Component Description
	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

Table 17 Security Assurance Requirements

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.

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	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.
ALC_CMS.1	
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.

Table 18 TOE Security Assurance Measures

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 the below Table 19:

Protocol	Key Exchange	Authentication	Cipher	Integrity
IKEv1	Group 14 (FFC MODP 2048) Group 19 (EC P-256) Group 20 (EC P-384)	RSA 2048 RSA 4096 ECDSA P-256 ECDSA P-384	AES-CBC-128 AES-CBC-192 AES-CBC-256 AES-GCM-128 AES-GCM-256	SHA-256 SHA-384
IKEv2	Group 14 (FFC MODP 2048) Group 19 (EC P-256) Group 20 (EC P-384)	RSA 2048 RSA 4096 ECDSA P-256 ECDSA P-384	AES-CBC-128 AES-CBC-192 AES-CBC-256 AES-GCM-128 AES-GCM-256	SHA-256 SHA-384
IPsec ESP	IKEv1 with optional: • Group 14 (FFC MODP 2048) • Group 19 (EC P-256) • Group 20 (EC P-384)	IKEv1	AES-CBC-128 AES-CBC-192 AES-CBC-256 AES-GCM-128 AES-GCM-256	HMAC-SHA-1-96 HMAC-SHA-256-128
	IKEv2 with optional: • Group 14 (FFC MODP 2048) • Group 19 (EC P-256) • Group 20 (EC P-384)	IKEv2	AES-CBC-128 AES-CBC-192 AES-CBC-256 AES-GCM-128 AES-GCM-256	HMAC-SHA-1-96 HMAC-SHA-256-128
SSHv2	ECDH-SHA2-nistp256 ECDH-SHA2-nistp384 ECDH-SHA2-nistp521	RSA 2048 RSA 3072 RSA 4096 ECDSA P-256 ECDSA P-384 ECDSA P-521	AES-CTR-128 AES-CTR-256 AES-CBC-128 AES-CBC-256	HMAC-SHA2-256 HMAC-SHA2-512

Table 19 Protocol Usage of Cryptographic Algorithms

Requirement	TSS Description
FAU_GEN.1, FAU_GEN.1/IPS, FAU_GEN.1/VPN, FAU_GEN.2	Junos OS creates and stores audit records for the following events (the details of content recorded for each audit event is detailed in Table 14, Table 15 and Table 16. Local auditing is implemented using syslog. In addition the management activities of TSF data, as defined under FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMG.1/IPS and FMT_SMF.1/VPN are recorded: 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, these audit settings must be configured as per the guidance document. 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. String-based detection through inspection of protocol fields is described in the IPS_SBD_EXT.1 TSS section. In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, and deleting keys): • PKID – certificate id of the associated certificate will be recorded when generating or deleting a key pair used for IPsec • Username and key type - The key type (rsa/ecdsa) and the username of the associated user will be recorded when importing an SSH user public key. For SSH (ephemeral) session keys the PID is used as the key reference to implicitly relate the key generation and key destruction audit events. The key destruction event is recorded as a session disconnect event. 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

Requirement	TSS Description
	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. The Wind River Linux host OS 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 the syslog utility is used to store the audit logs locally, and optionally to send them to one or more syslog servers in real time via Netconf over SSH. Local audit logs are stored in /var/log/ in the underlying filesystem in a persistent format. Only a Security Administrator can read or clear/delete active and archived log files through the CLI interface or through direct access to the filesystem having first authenticated as a Security Administrator. The local logs are automatically overwritten according to configurable limits on storage volume. The default maximum size is 1 MB, which can be modified by the user, using the “`size`” argument for the “`set system syslog file <filename> archive`” 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 the /var filesystem, which is 3.9 GB for NFX platforms. However, when this 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, 3072, 4096 bit) • ECC (P-256, P-384, P-521) • FFC (2048-bit MODP). Usage of the keys in protocols is specified in Table 19.

<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 186-5 for IKE with IPSec. The TOE complies with NIST SP 800-56A regarding asymmetric FFC 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 186-5 Appendix A.1 (for RSA schemes) and A.2 (for ECDSA schemes).
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>Key establishment is performed in accordance with NIST Special Publication 800-56A Revision 3 for ECC Schemes, for SSH and IPsec communications. The TOE also implements FFC key establishment in accordance with NIST Special Publication 800-56A Revision 3, using the modulus and generator specified by Section 3 of RFC3526. Usage of key agreement in protocols is specified in Table 19.</td></tr>
<tr><td>FCS_CKM.4</td><td>Table 22 of the Security Target lists all relevant keys consistent with the functions carried out by the TOE, as well as their origin, storage location, situations in which keys are destroyed and key destruction method used. The TOE stores all keys in plaintext form in either volatile or non-volatile memory. 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 19. 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 19. 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>512</th></tr>
<tr><td>Key Length</td><td>160 bits</td><td>256 bits</td><td>512 bits</td></tr>
<tr><td>Hash function</td><td>SHA-1</td><td>SHA-256</td><td>SHA-512</td></tr>
<tr><td>Block Size</td><td>512 bits</td><td>512 bits</td><td>1024 bits</td></tr>
<tr><td>Output MAC</td><td>160 bits</td><td>256 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, 3072 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 with no configuration necessary.
By default, the TOE denies all traffic through the NFX series device. In fact, an implicit default security policy exists that denies all packets. The administrator can change this default 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: bypass, discard, protect and 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 for both inbound as well as outbound packets: • Bypass – The `permit` action only directs traffic traversing the device through the stateful firewall inspection, but not through or around the IPsec VPN tunnel. The latter will be based on a combination of route lookup and `permit` policy inspection. Security policies with a `permit` action combined with a non-tunnel route will act as ‘bypass’ rules. • Discard – The Deny option inspects and drops all packets matching the rule. • Protect – Security policies with a `permit` action combined with a tunnel-based route directing the traffic via the virtual ‘st0’ tunnel interface will act as ‘protect’ rules. • 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, AES-GCM-256, AES-CBC-128, AES-CBC-192 or AES-CBC-256 using HMAC-SHA-1 and HMAC SHA-256 for ESP protection. For the mentioned HMAC algorithms, the TOE supports and only implements truncated outputs, more specifically, HMAC-SHA-1-96, truncating the normally 160-bit output to 96 bits, and HMAC-SHA-256-128, truncating the normally 256-bit output to 128 bits. Both of these are the only configurable options under the `security ipsec proposal` configuration hierarchy. 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 RFC 7296 (with no support for NAT traversal) and RFC 4868 for hash functions. IKEv1 aggressive mode is configurable, but out of scope, and hence not to be used

Requirement	TSS Description

	when the TOE is deployed in a CC configuration, wherein main mode must be the only configured option.
	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 86,400 seconds i.e. 0.05 to 24 hours),
	• IKEv1 Phase 2 SA in terms of length of time (180 to 28,800 seconds i.e. 0.05 to 8 hours)
	• IKEv2 Child SA lifetimes in terms of kilobytes (64 to 4,294,967,294) and length of time (180 to 28,800 seconds i.e. 0.05 to 8 hours).
	The TOE implements the following CLI commands to configure the Phase 1 lifetime in seconds:
	`set security ike proposal <name> lifetime-seconds <seconds>`
	Phase 2/Child SA lifetime is configured in seconds using the following command:
	`set security ipsec proposal <name> lifetime-seconds <seconds>`
	Child SA lifetime is configured in kilobytes using the following command:
	`set security ipsec proposal <name> lifetime-kilobytes <kb>`
	The TOE supports Diffie-Hellman Groups 14, 19, and 20. When the TOE receives an IKE proposal, it will match the DH group with the DH group configured on the TOE (one out of DH Groups 14, 19, or 20) and the negotiation will only succeed if there is an exact match. A DH group must be configured on the TOE in the evaluated configuration (FIPS mode), in the absence of which, an error is generated on committing the configuration.
	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), 256 bits (for DH Group 19) or 384 bits (for DH Group 20).
	The TOE supports both RSA and ECDSA for use with X.509v3 certificates that conform to RFC 4945 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 commit fails.
	The TOE uses X.509v3 certificates with RSA and ECDSA as defined in RFC 4945. Certificate Request Messages are generated in accordance with RFC 2986 when validating certificates for IPsec connections.
	The TOE supports the use of an IP address, Fully Qualified Domain Name
	(FQDN) or user FQDN in the SAN field of the certificate as reference identifiers, along with the Distinguished Name (DN). CN reference identifiers

Requirement	TSS Description
	are not supported and contents of the field are disregarded, unless the DN reference identifier is being used. The TOE validates reference identifiers by comparing the configured identifier against the SAN or DN field of the presented peer certificate depending on the configured identifier. The connection is only accepted on an exact match between the two. The use of certificates is described in FIA_X509_EXT.1/Rev and FIA_X509_EXT.3.
FCS_RBG_EXT.1	Junos OS performs random bit generation in accordance with NIST Special Publication 800-90A using HMAC_DRBG, SHA-256. The kernel DRBG for the NFX TOE does not require any configuration and requires 448 bits of credited entropy from the primary source i.e. bits 2-9 of the timestamp of the software interrupts associated with the clock0 (RANDOM_SWI_CLOCK0) to be fully seeded, with the source supplying a min-entropy of at least 0.83 per byte.
FCS_SSH_EXT.1/ FCS_SSHS_EXT.1	The TOE acts as 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 TOE also acts as an SSH server also supports Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. Remote administrators of Junos OS initiate communication to the Junos CLI through the SSH tunnel created by the SSH session. 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, 4256, 4344, 5656,6668 and 8332. Junos OS provides assured identification of the SSH client through public key authentication, keyboard-interactive and password-based authentication by administrative users (Security Administrator) for SSH connections. Host Keys: By default, 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. Using configuration via the CLI as per the guidance document, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, ssh-rsa, rsa-sha2-256 and rsa-sha2-512 are also configured as supported host key algorithms. This key is randomly generated to be unique to each TOE instance. The TOE presents the client with its public key and the client matches this key against its known_hosts list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different (per the SSHv2 protocol). Confidentiality: The TOE does not accept the “none” cipher and supports AES-CBC-128, AES-CBC-256, AES-CTR-128 and AES-CTR-256 encryption algorithms for protection of data over SSH and uses keys generated in accordance with “ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256,

Requirement	TSS Description
	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 is configured using a time or data-based threshold. The data-limit can be set between 51200 bytes (~0.00005 GB) to 4294967295 bytes (~4 GB) and the time-limit can be set between 1 to 1440 minutes (~0.016 to 24 hours). The TOE will rekey based on whichever limit is reached first. Authentication: 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: ssh-rsa, rsa-sha2-256, rsa-sha2-512, 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. Password Authentication Method: The TOE supports and implements password authentication as described in the FIA_UIA_EXT.1 TSS section. Keyboard-interactive Authentication method: The TOE supports keyboard-interactive authentication. Providing multifactor authentication mechanism would require the use of an external AAA server, which is outside the CC scope, as a result of which the keyboard-interactive authentication method works similarly to the password-based method in the evaluated configuration. 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. ECDH Key Exchange: The supported key exchange methods specified are ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521. The client matches the key against its known_hosts list of keys. For the mentioned ECDH key exchange algorithms, the TOE uses the SSH KDF as defined in RFC 5656 (Section 4). Data Integrity Algorithms: hmac-sha2-256 and hmac-sha2-512 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

Requirement	TSS Description
	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 • FIPS self-tests, firmware, kernel and host OS integrity tests are executed • Loading and initialization of the FreeBSD Kernel • 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 the power-up tests are successful, all of the components on the appliance are fully initialized 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. In case there are situations where the traffic exceeds the amount of traffic the TOE can handle, it starts by prioritizing traffic belonging to existing sessions and drops traffic for which no existing sessions can be identified. If the volume still exceeds the TOE's capacity, it starts dropping packets at the INQ (queue of ingress packets) level itself. In this way, the TOE ensures that even in situations of high ingress volume exceeding the TOE's capacity, no traffic is allowed through without being subject to the stateful filtering rules, preventing traffic that shouldn't pass from being allowed. 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, thereby ensuring that packets are never let through without applying the configured ruleset in the event of a component failure of any kind. 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. In cases of conflicting rules being defined, the one first in order will be enforced. By default, the TOE behavior is to deny packets when there is no rule match. 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 • Security Policy Module • Session Lookup 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 Security Policy module examines traffic passing through the TOE (via Session Setup module) and determines if the traffic can pass based on administrator-configured access policies. The Security Policy module is the core of the firewall and IPS functionalities in the TOE: It is the policy enforcement engine that fulfills the security requirements for the user. The Security Policy module will deny packets when there is no policy match unless another policy allows the traffic. 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. If the Security Policy module determines that a packet is permitted by policy, it is passed to the Session Lookup module to perform lookups in the session table.

Requirement	TSS Description
	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 other modules. Eventually if the packet is not destined for the TOE, the Network interface will pass the traffic out of the appliance. 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 2463 ICMPv6: Type, Code • RFC 791 (IPv4): Source address, Destination Address, Transport Layer Protocol • RFC 8200 (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 supports the entire list of IPv4 protocols as per the RFC values in the table. However, for IPv6, protocol IDs 43,44 and 60 are not supported by the TOE, and their packets are hence dropped without logging before being subject to policy matching. 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 'ge' (gigabit ethernet) interfaces and the 'st0' VPN tunnel virtual interface are the ones subject to the stateful packet filtering policy, with each of these interfaces assigned to a 'security zone'. The TOE then enforces stateful

Requirement	TSS Description
	packet filtering through security policies, which are defined between pairs of security zones (by defining a ‘from zone’ and a ‘to zone’). The TOE can also enforce stateful filtering via firewall filters, which are directly applied to the network interfaces for ingress or egress traffic. 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 For TCP, the TOE tracks the full three-way handshake to establish stateful sessions: 1. SYN : Initial connection request from client. 2. SYN-ACK : Response from server. 3. ACK : Final acknowledgment from client. A TCP session is established in the TOE only after this handshake is successfully observed. The TOE validates sequence numbers and ensures both endpoints have agreed to the connection. TCP session maintenance includes: • Monitoring TCP flags (e.g., PSH, ACK, FIN, RST) to detect session teardown (a FIN or RST exchange) and initiate session removal. • Enforcing the correct sequence and state transitions (e.g., from SYN_SENT to ESTABLISHED). • Maintaining per-flow state in the session table. • Enforcing session removal based on idle timeouts (1800 seconds by default). While UDP is stateless at the protocol level, the TOE treats UDP traffic in a session-like manner: • A session is created upon receipt of a valid UDP packet that matches a permitted policy. • The TOE creates a unidirectional or bidirectional flow entry in the session table based on source/destination IPs and ports. • There is no handshake; the session is considered active as long as matching UDP traffic continues. UDP session maintenance includes: • Tracking each UDP flow’s parameters (source IP/port, destination IP/port). • Applying idle timeouts (60 seconds by default). • Removing the session after no matching traffic is observed for the configured timeout duration. ICMP is not connection-oriented, but the TOE implements session tracking for ICMP messages such as Echo Request / Echo Reply. ICMP session establishment:

Requirement	TSS Description
	• When an ICMP request is observed and permitted by policy, the TOE creates a temporary session entry. • This session is considered "established" for the purpose of tracking the reply and enforcing policy symmetry. ICMP Session maintenance: • The session exists briefly • The session allows the associated ICMP reply message (e.g., Echo Reply). • Once the timeout (6 seconds by default) expires or the reply is seen, the session is removed. Session removal becomes effective immediately after expected flow completion or timeout expiry, following which the session entry is removed from the session table before any subsequent packets are evaluated. Specific timeouts for each of the three above-mentioned protocols can also be manually configured by defining policy applications using the `applications application <policy application name> term <term name>` configuration hierarchy, by setting a value for the `inactivity-timeout` variable. The TOE supports FTP (RFC 959) to dynamically establish sessions allowing network traffic according to Administrator rules. Stateful filtering policies are configured with port ranges to handle dynamic FTP sessions. Since FTP utilizes TCP at the transport layer, the same behavior applies in terms of session establishment and removal. Session events will be logged in accordance with ‘log’ operations defined in the rules. Source and destination addresses, source and destination ports, transport layer protocol, and TOE Interface are recorded in each log record. The TOE enforces the following default reject rules with logging on all network traffic after the corresponding security mechanisms are enabled: • 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; • 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;

Requirement	TSS Description
	• with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; • packets are checked for validity. “Invalid fragments” are those that violate these rules: ○ No overlap ○ The total fragments in one packet should not be more than 62 pieces ○ The total length of merged fragments should not larger than 64k ○ All fragments in one packet should arrive in 2 seconds ○ The total queued fragments has limitation ○ The total number of concurrent fragment processing for different packet has limitations 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 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’ parameter is 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 SSH access. The retry-options are applied following the first failed login attempt for a given username. The ‘tries-before-disconnect’ sets the maximum number of times (2-10) 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). A locked out user can also be manually unlocked before the lockout period elapses using the `clear system login lockout user <user>` command. 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 can attempt authentication via remote access after the lockout period elapses.
FIA_PMG_EXT.1	Authentication data for fixed password authentication is a case-sensitive, value containing a combination of alphanumeric and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “’”, ”’”, ”+”, “,”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “?”, “[”, “\”, “]”, “_”, “`”, “{”, “|”, “}”, “~”. The

Requirement	TSS Description
	password complexity requirements must be manually configured by the administrator via the CLI using the ‘`set system login password`’ configuration hierarchy as per following specifications: - Minimum length – Range: 6 to 20 characters (default value set to 10 in FIPS mode if not explicitly configured). A minimum length configuration of 15 characters is recommended when used in CC configuration. - Must contain at least one character each from the different character sets (uppercase, lowercase, numeric and special characters). - Maximum length (optional) – Range: 20 through 128 characters (default value set to 128 if not explicitly configured).
FIA_UIA_EXT.1, FIA_UAU_EXT.1, FIA_UAU.7	Junos users are configured under the the `system login user` configuration hierarchy 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 the above-mentioned configuration hierarchy, 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. This ‘login’ process is either accessed through 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 (both remote and local), 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, following which either access to the CLI is provided (correct password) or the password prompt is presented again (incorrect password). 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

Requirement	TSS Description
	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 and public keys for remote SSH and just passwords for local serial console) 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 (for remote management terminals).
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. It does not use certificates for trusted updates/executable code integrity, DTLS/TLS (not supported) or OCSP (CRL used for revocation checking), thereby not requiring the corresponding extendedKeyUsage fields mentioned in FIA_X509_EXT.1.1/Rev. When certificates are used for authentication in IPsec, the certificate validity and revocation 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 validity time period specified in the certificate. When the TOE is configured to perform a revocation check using CRL (as specified in RFC 5280 Section 6.3) and 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. Revocation checking using CRL is performed for the leaf and ICA certificates. The TOE validates a certificate path by building a chain of (at least 3) certificates based upon issuer and subject linkage, validating each according to 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. 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. If the TSF cannot establish a connection to determine the validity of a certificate, the TOE takes the action configured by the administrator. In FIPS mode, “disable on-download-failure” may be set for a CA to allow connections to be established when CRLs could not be retrieved . Otherwise,

Requirement	TSS Description
	connections involving a CA for which the specified CRL could not be retrieved are rejected.
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 • One of the following SAN types: ○ 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	The TOE allows administrators to modify the behaviour of the following administrative functions using CLI commands: • transmission of audit data to an external IT entity – using the '`system services netconf`' configuration hierarchy. • handling of local audit data– using the '`system syslog`' configuration hierarchy.
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 i.e. start/stop the following functions by issuing '`delete`' commands via the CLI to remove configuration for the respective services: • the trusted communication channel to the external syslog server (netconf over SSH) • the trusted path for remote Administrative sessions (SSH)
FMT_MTD.1/CoreData	The Administrator (security-admin) is the only role with the ability to manage the TSF data, which includes the trust store the TOE implements to support handling of X.509v3 certificates. The Administrator performs management functions via a specialized out-of-band management interface. No administrative functions are accessible prior to logging in.
FMT_MTD.1/CryptoKeys	The Security Administrator has the capability to perform the following operations related to SSH public keys using the '`system login user`' configuration hierarchy: • import • delete The Security Administrator also has the capability to perform the following operations related to keypairs used for IPsec using the '`security pki`' hierarchy:

Requirement	TSS Description
	• generate • delete • import (by importing signed CSR responses)
FMT_SMF.1, 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 management functions necessary to manage Junos OS in accordance with the requirements, all of which can be performed using both the directly connected serial console local interface as well as the SSH remote administrative interface. The Security Administrator has the capability to: ○ Ability to administer the TOE remotely; ○ Ability to configure the access banner; ○ Ability to configure the remote session inactivity time before session termination; ○ Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates; ○ Ability to start and stop services; ○ Ability to configure local audit behavior; ○ Ability to modify the behavior of the transmission of audit data to an external IT entity; ○ Ability to manage the cryptographic keys; ○ Ability to manage the cryptographic functionality; ○ Ability to configure thresholds for SSH rekeying; ○ Ability to configure the lifetime for IPsec SAs; ○ 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 administer the TOE locally; ○ Ability to configure the local session inactivity time before session termination or locking; ○ Ability to configure the authentication failure parameters for FIA_AFL.1; ○ Ability to manage the trusted public keys database; ○ 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; ○ 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
	o Destination IP addresses (host address and network address); o Source port (TCP and UDP); o Destination port (TCP and UDP); o Protocol (IPv4 and IPv6) o ICMP type and code o Update (import) IPS signatures; o Create custom IPS signatures; o Configure anomaly detection; o Enable and disable actions to be taken when signature or anomaly matches are detected; o Modify thresholds that trigger IPS reactions; o Modify the duration of traffic blocking actions; o Modify the known-good and known-bad lists (of IP addresses or address ranges); o 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). The individual username is recorded whenever an audit log is generated for administrator actions. The TOE implements system logging locally as described in the FAU_STG_EXT.1 TSS section.
FMT_SMR.2	The TOE implements a Security Administrator role 'security-admin'. It is the only role authorized to administer the TOE. Each user assigned to the Security Administrator role gains access to the full CLI. Each human security-admin is identified and authenticated with a username and password and assigned a Security Administrator role upon successful authentication. The role assignment remains until the session is terminated. The TOE also supports a self-explanatory 'ready-only' role, which only has view permissions for the TOE configuration and operational state (`show` commands).
FPT_APW_EXT.1	The user passwords are stored in obfuscated form using the Modular Crypt Format (MCF), utilizing sha-256 or sha-512 depending on the configured algorithm, with a 16 byte salt value and 5000 rounds of hashing, and are displayed in the same form when the configuration is viewed using the 'show' command.
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. • File integrity test –verifies integrity of all mounted signed packages, to assert that system files have not been tampered with. This includes RSA3072/SHA256 signature verification of the WRL host OS

Requirement	TSS Description
	and initrd (initial ramdisk) file, ECDSA256/SHA256 signature verification of the FreeBSD packages and ECDSA256/SHA256 signature verification of packages constituting the Junos firmware. To test the integrity of the firmware, the fingerprints of the executables and other immutable files are regenerated and validated against the fingerprints contained in the manifest file. • Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys and iked credentials, such as CAs, certificates, and various keys. • Simulated veriexec error – verifies that the veriexec file integrity enforcement framework is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real, which triggers an intentional failure and verifies that it gets detected. • Crypto self-tests (Kernel, libmd, OpenSSL, QuickSec and QAT libraries) for SSH and IPsec – verifies correct output from known answer tests for all cryptographic algorithms. • Noise source health tests to verify the correct operation of the noise source. Tests include a repetitive count test and an adaptive proportion test. 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. The self-tests ensure that only authorized untampered executables are allowed to run, and that the noise source and cryptographic algorithms are operating as expected, thus ensuring the correct operation of the TOE. In the event of a transiently corrupt state or failure condition within the TOE, 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, generates a core dump file and reboots. No command line input or traffic to any interface is processed. If the failure persists on rebooting, the TOE ends up in a continuous core dump and reboot cycle until the administrator intervenes.
FPT_SKP_EXT.1	Storage of symmetric keys, and private keys is described in Table 22. 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 (out of scope as mentioned in Section 1.3.3).
FPT_STM_EXT.1	All events recorded by the syslog daemon 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, user lockouts and IPsec/SSH rekey thresholds. The Wind River Linux host OS 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.

Requirement	TSS Description
	The time is kept reliable by the administrators setting the accurate time during the first-time installation and at the start of every day thereafter.
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. 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 installation fails for any reason an 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 Wind River Linux the Junos VM installation will proceed. The Junos OS kernel maintains a set of fingerprints for executable files and other files which should be immutable. 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 Administrators configure the TOE so that a user session, remote or local, 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 terminated. The Security Administrator configures this inactivity timer, which is enforced for both local and remote administrative sessions, after which the session will be logged out, exiting the display device to the login prompt.
FTA_SSL.4	User sessions (local and remote) can be terminated by users. The administrative user can log out of the existing session by issuing the `quit` or

Requirement	TSS Description
	`exit` commands to exit the CLI admin session. 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 provides warnings against unauthorized access to the TOE as well as any other information that the Security Administrator wishes to communicate. The same configured banner is shown at the local serial console and during remote access sessions using SSH.
FTP_ITC.1	Junos OS supports Trusted Channels using SSHv2 protocol, with the TOE acting as an SSH server, which ensures the confidentiality and integrity of communication with the remote audit server. The audit server endpoint is assuredly identified using the presented password or public key. 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. The TOE can act as both initiator and responder. NFX350 supports numerous routing standards as well as IPSec protocols. These functions are managed through the Junos OS software, either from a connected console on the management interface or via a network connection. The TOE supports the use of an IP address, Fully Qualified Domain Name (FQDN) or user FQDN in the SAN field of the certificate as reference identifiers, along with the Distinguished Name (DN). CN reference identifiers are not supported and contents of the field are disregarded, unless the DN reference identifier is being used. The TOE validates reference identifiers by comparing the configured identifier against the SAN or DN field of the presented peer certificate depending on the configured identifier. The connection is only accepted on an exact match between the two. The TOE supports AES-GCM-128 and AES-GCM-256, and AES-CBC-128, AES-CBC-192 or AES-CBC-256 using HMAC-SHA-1-96 and HMAC SHA-256-128 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 RFC 7296 (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	Junos OS supports Trusted Paths using SSHv2 protocol for remote administration, which ensures the confidentiality and integrity of user sessions. 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

Requirement	TSS Description
	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 are defined to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic. An IDP policy is made up of rule bases, and each rule base contains a set of rules that specify rule parameters, such as traffic match conditions, action, and logging requirements. IDP policies are then associated to firewall policies. IDP is 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 is 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 is 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. • 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. 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.

Requirement	TSS Description
	The TOE is capable of inspecting all traffic passing through the TOE's Gigabit Ethernet (ge) interfaces (inline mode). Each of these interfaces types can be assigned to Zones on which firewall and IDP policies are predicated. The dedicated out-of-band management 'fxp0' interface (control plane), however, cannot be assigned to such security zones, are not subject to intrusion prevention and detection functions and are hence logically distinct from the sensor interfaces (data plane). The TOE supports the definition of known-good and known-bad lists of source and/or destination addresses at the firewall rule level. These can be single IPs or a range of IPs. Address ranges are defined by creating address book entries and attaching them to firewall policies. Addresses in known-good lists would be combined with a 'permit' action in the firewall policies, while those in the known-bad lists with a 'deny' action. Security administrators are the only ones capable of defining IPS policy elements for the known-good and known-bad lists. 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; payload length; next header; hop limit; source address; destination address; routing header; traffic class; and flow label; • ICMPv4: Type; Code; Header Checksum; ID; and sequence number. • 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 any of the 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 supports stateful signature-based attack detection defined as Attack Objects called IDP custom attacks. They use context-based matching to match regular expressions in specific locations where they occur. They can be composed of multiple signatures and protocol anomalies, including logical expressions between signatures for compound matching. These custom

Requirement	TSS Description

attacks are comprised of the 'context' to look for the pattern (within a packet, within a stream etc.), the 'direction' in which to detect the pattern (any, client to server, or server to client) and the protocol header and field to be inspected. The custom attack is then bound to an IDP policy which specifies the matching criteria in terms of source and destination, and the action to be taken.

Signature-based IPS policies can be associated with the following actions:

- No action i.e. allow
- Drop packet/connection
- Send TCP RST to the source
- Log

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 SMTP states. Alternative, administrators can 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:

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

The default action for the above screens is to drop the packets.

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 syn";

Requirement	TSS Description
	- 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 a screen under the ‘screen ids-option’ hierarchy and defining the source, destination and alarm thresholds under ‘tcp syn-flood’. (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 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. 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 performed on signature triggering (allow or block/drop traffic). Anomaly signatures based on frequency characteristics are implemented by defining standard firewall filters for specific source/destination address, source/destination port and protocol combinations, and configuring the ‘count’ parameter in the action to maintain a counter for hits. Anomaly signatures based on threshold characteristics are implemented by configuring probes under the ‘services rpm’ (Real-time performance monitoring) configuration hierarchy and defining the target address, type of threshold to monitor and corresponding traps to be sent to the logging utility when the threshold is met or exceeded. Anomaly-based IPS policies can be associated with the following actions: - No action i.e. allow - Drop packet/connection - Log

Table 20 TOE Summary Specifications

6.1 CAVP Algorithm Certificate Details

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

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of [2048 bits or greater] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;	Junos 23.4R1 OpenSSL	RSA KeyGen (186-5) Modulo: 2048, 3072, 4096	A5151
	ECC schemes using 'NIST curves' [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;	Junos 23.4R1 OpenSSL	ECDSA KeyGen (186-5) Curve: P-256, P-384, P-521 ECDSA KeyVer (186-5) Curve: P-256, P-384, P-521	A5151
	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].	N/A	FFC safe prime groups testing is expected to be performed in conjunction with FCS_CKM.2.1	FFC safe prime groups testing is expected to be performed in conjunction with FCS_CKM.2.1
FCS_CKM.1/IKE	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes	Junos 23.4R1 OpenSSL	RSA KeyGen (186-5) Modulo: 2048, 4096	A5151
	FIPS PUB 186-5, "Digital Signature Standard (DSS)," Appendix B.4 for ECDSA schemes, and implementing "NIST curves" P-384 and [P-256]	Junos 23.4R1 OpenSSL	ECDSA KeyGen (186-5) Curve: P-256, P-384 ECDSA KeyVer (186-5) Curve: P-256, P-384	A5151
	FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision	N/A	FFC safe prime groups testing is expected to be performed in conjunction with FCS_CKM.2.1	FFC safe prime groups testing is

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
	3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526]			expected to be performed in conjunction with FCS_CKM.2.1
FCS_CKM.2	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”	Junos 23.4R1 OpenSSL	KAS-ECC-SSC Sp800-56Ar3 Domain Parameter Generation Methods: P-256, P-384, P-521	A5151
		Junos 23.4R1 QAT	KAS-ECC-SSC Sp800-56Ar3 Domain Parameter Generation Methods: P-256, P-384	A5157
	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 [groups listed in RFC 3526]	Junos 23.4R1 OpenSSL	KAS-FFC-SSC Sp800-56Ar3 Domain Parameter Generation Methods: modp-2048	A5151 Also tested by the lab against known-good implementation.
		Junos 23.4R1 QAT	KAS-FFC-SSC Sp800-56Ar3 Domain Parameter Generation Methods: modp-2048	A5157 Also tested by the lab against known-good implementation.
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR, GCM] mode and cryptographic key sizes [128 bits, 192 bits, 256 bits]. 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].	Junos 23.4R1 OpenSSL	AES-CBC Direction: Decrypt, Encrypt Key Length: 128, 192, 256 AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 AES-CTR Direction: Decrypt, Encrypt Key Length: 128, 256	A5151
		Junos 23.4R1 Quicksec	AES-CBC Direction: Decrypt, Encrypt	A5152

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
			Key Length: 128, 192, 256 AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256	
		Junos 23.4R1 QAT	AES-CBC Direction: Decrypt, Encrypt Key Length: 128, 192, 256 AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256	A5157
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	Junos 23.4R1 OpenSSL	RSA SigGen (FIPS186-5) Signature Type: pkcs1v1.5 Modulo: 2048, 3072, 4096 RSA SigVer (FIPS186-5) Signature Type: pkcs1v1.5 Modulo: 2048, 3072, 4096	A5151
		Junos 23.4R1 Quicksec	RSA SigGen (FIPS186-5) Signature Type: PKCS 1.5 Modulo: 2048, 4096 RSA SigVer (FIPS186-5) Signature Type: PKCS 1.5 Modulo: 2048, 4096	A5152
	For ECDSA schemes implementing [selection: P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST Recommended” curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.	Junos 23.4R1 OpenSSL	ECDSA SigGen (FIPS186-5) Curve: P-256, P-384, P-521 ECDSA SigVer (FIPS186-5) Curve: P-256, P-384, P-521	A5151
		Junos 23.4R1 QAT	ECDSA SigGen (FIPS186-5) Curve: P-256, P-384 ECDSA SigVer (FIPS186-5) Curve: P-256, P-384	A5157
FCS_COP.1/ Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits	Junos 23.4R1 LibMD	SHA-1 SHA2-256	A5150
		Junos 23.4R1 OpenSSL		A5151

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
	that meet the following: ISO/IEC 10118-3:2004.	Junos 23.4R1 Quicksec	SHA2-384 SHA2-512	A5152
FCS_COP.1/ KeyedHash	[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512] and cryptographic key sizes [160, 256, and 512 bits] and message digest sizes [160, 256, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.	Junos 23.4R1 OpenSSL	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-512	A5151
		Junos 23.4R1 QAT		A5157
FCS_RBG_EXT.1	HMAC_DRBG [SHA-256] in accordance with ISO/IEC 18031:2011	Junos 23.4R1 Kernel	HMAC DRBG	A5149

Table 21 CAVP Algorithm Certificate References

6.2 Cryptographic Key Descriptions

The table below describes the keys provided by the TOE.

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. RSA (2048, 3072 and 4096) and EC (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 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)

Keys/CSPs	Purpose	Method of Storage	Storage Location	Method of Zeroization
SSH Session Key	Session keys used with SSH, AES (CTR/CBC) 128, 256, hmac-sha2-256 or hmac-sha2-512 key (256 or 512), DH Private Key (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 (SHA-256 crypt and SHA-512 crypt)	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 used in IKE. RSA 2048, RSA 4096, ECDSA P-256, ECDSA P-384.	Plaintext	Disk (mapped to SDD)/Memory	'clear security IKE security-association' command or reboot the box. Private keys stored in flash are not zeroized unless an explicit "request system zeroize" is executed.
IKE-SKEYID	IKE master secret used to derive IKE and IPsec ESP session keys	Plaintext	Memory	'clear security IKE security-association' command or reboot the box

Keys/CSPs	Purpose	Method of Storage	Storage Location	Method of Zeroization
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, ECDH P-384 or ECDH P-521	Plaintext	Memory	‘clear security IKE security-association’ command or reboot the box.
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)

Table 22 Key Descriptions

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 23 – 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