Berryville Holdings, LLC Warden v1.2 Security Target

Version 0.6
07/03/2025

Prepared for:

Berryville Holdings, LLC

2465 Centerville Road,
Herndon, VA 20171

Prepared By:

www.gossamersec.com

1. SECURITY TARGET INTRODUCTION 4
1.1 SECURITY TARGET REFERENCE 4
1.2 TOE REFERENCE 4
1.3 TOE OVERVIEW 5
1.4 TOE DESCRIPTION 5
1.4.1 TOE Architecture 5
1.4.2 TOE Documentation 9
2. CONFORMANCE CLAIMS 10
2.1 CONFORMANCE RATIONALE 12
3. SECURITY OBJECTIVES 13
3.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 13
4. EXTENDED COMPONENTS DEFINITION 15
5. SECURITY REQUIREMENTS 16
5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 16
5.1.1 Security audit (FAU) 18
5.1.2 Cryptographic support (FCS) 22
5.1.3 User data protection (FDP) 26
5.1.4 Firewall (FFW) 26
5.1.5 Identification and authentication (FIA) 28
5.1.6 Security management (FMT) 30
5.1.7 Packet Filtering (FPF) 32
5.1.8 Protection of the TSF (FPT) 32
5.1.9 TOE access (FTA) 33
5.1.10 Trusted path/channels (FTP) 34
5.1.11 Intrusion Prevention (IPS) 35
5.2 TOE SECURITY ASSURANCE REQUIREMENTS 37
5.2.1 Development (ADV) 37
5.2.2 Guidance documents (AGD) 38
5.2.3 Life-cycle support (ALC) 39
5.2.4 Tests (ATE) 39
5.2.5 Vulnerability assessment (AVA) 39
6. TOE SUMMARY SPECIFICATION 41
6.1 SECURITY AUDIT 41
6.2 CRYPTOGRAPHIC SUPPORT 42
6.3 USER DATA PROTECTION 45
6.4 FIREWALL 45
6.5 IDENTIFICATION AND AUTHENTICATION 48
6.6 SECURITY MANAGEMENT 48
6.7 PACKET FILTERING 50
6.8 PROTECTION OF THE TSF 51
6.9 TOE ACCESS 52
6.10 TRUSTED PATH/CHANNELS 52
6.11 INTRUSION PREVENTION 52

LIST OF TABLES

Table 1 IT Environment Components 6
Table 2 Technical Decisions 12
Table 3 TOE Security Functional Components 17

Table 4 Auditable Events 20
Table 5 IPS Auditable Events 21
Table 6 VPN Auditable Events 22
Table 7 Assurance Components 37
Table 8 CAVP Certs 42
Table 9 Key Usage Schemes 43
Table 10 Key Destruction 43
Table 11 Protocols & Fields Filtered by the TOE 46
Table 12 Connection Tracking Fields 47

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Berryville Warden provided by Berryville Holdings, LLC.. The TOE is being evaluated as a network device.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration may be indicated by a parenthetical number placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement. Alternately, a usually descriptive textual extension may be added after a slash (/) character to identify a specific iteration. For example, iterations of a requirement such as FCS_COP.1 might be identified as FCS_COP.1/HASH and FCS_COP.1/CRYPT.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – Berryville Holdings, LLC Warden v1.2 Security Target

ST Version – Version 0.6

ST Date – 07/03/2025

1.2 TOE Reference

TOE Identification – Berryville Holdings, LLC Warden v1.2

TOE Developer – Berryville Holdings, LLC.

Evaluation Sponsor – Berryville Holdings, LLC.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Berryville Warden v1.2. The TOE is a virtual networking device that provides secure remote administration (through SSH), VPN gateway functionality (using IKE/IPSec), stateful firewalling, and an Intrusion Prevention System (IPS).

1.4 TOE Description

The TOE consists of a single instance of virtual Berryville device running the Warden software v1.2 on a physical device running Ubuntu 22.04's KVM hypervisor.

The TOE is assumed to be installed and operated within a physically protected environment, administered by trusted and trained administrators. The TOE can be remotely administered via SSH or via a local console.

The VM leverages the physical hardware's ethernet interface or additional USB interfaces to create virtual interfaces. The TOE is able to filter connections to/from external IT entities using its stateful firewall/IP traffic filtering capabilities. In addition to packet filtering, the TOE can also provide Intrusion prevention via anomaly or signature-based detection.

The TOE provides VPN gateway capabilities, allowing the Engine to use IKE/IPsec to protect traffic exchanged with remote peer gateways (for a site-to-site VPN configuration) and with VPN clients. Site-to-site configuration can be used to protect data between the TOE and a remote syslog server or NTP server.

1.4.1 TOE Architecture

The TOE is a single virtual network device running the Berryville Warden v1.2 software that operates independently (i.e., it is not a distributed TOE) and communicates with entities in its operational environment including remote administrators (who securely connect to the TOE's administrative CLI through SSH), NTP servers (tunneled within IPsec for security), syslog/audit servers, and with IPsec peers (gateways).

The physical characteristics of the TOE platform are the following:

- Ubuntu 22.04 (Linux 5.15) KVM Hypervisor running on 11th Gen Intel(R) Core (TM) i7-1165G7 (Tiger Lake)

The TOE is deployed in an environment that includes the IT components illustrated in the following figure. The TOE itself is delivered as a virtual image to be installed on the KVM hypervisor. The administrator of the TOE may verify the TOE software and, if necessary, download and install the correct version.

The physical boundary of the TOE is illustrated below. Non-TOE components are summarized in **Table 1 IT Environment Components**. The TOE implements IPsec as a VPN Gateway and SSH Server for secure connectivity to the components of the environment.

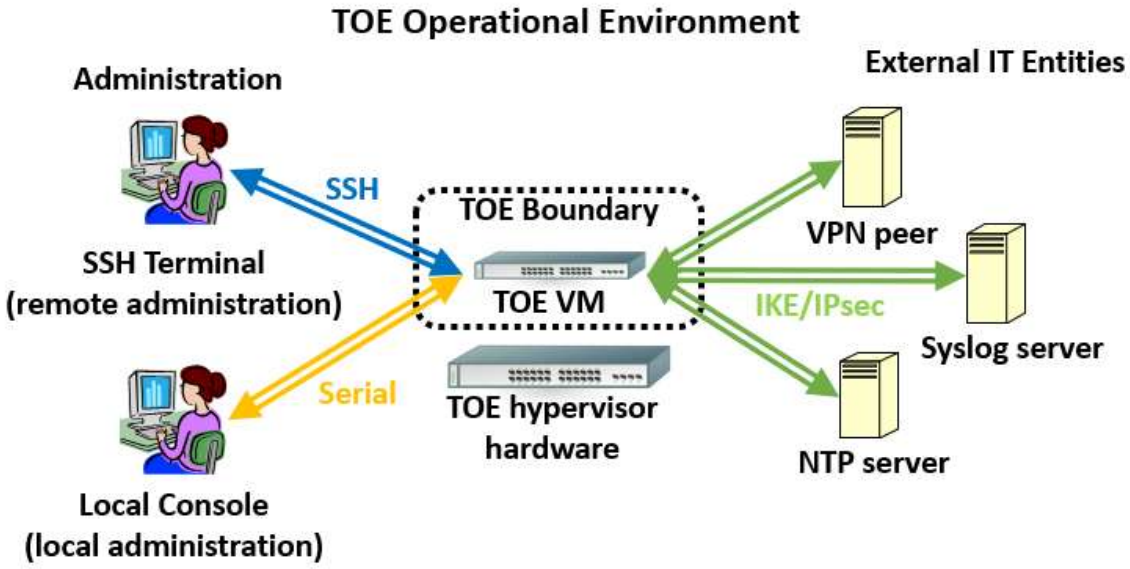

Figure 1 - TOE and TOE Operational Environment

The environmental components described in the following table are required to operate the TOE in the evaluated configuration.

Component	Description
VPN Peer	A peer server that supports IPsec to communicate with the TOE so the TOE can provide VPN Gateway functionality,
Audit (syslog) server	The audit server supports syslog messages over IPsec via a site-to-site connection to receive the audit logs from the TOE. The audit data is stored in the remote audit server for redundancy purposes.
NTP Server	The NTP server supports NTP messages over IPsec via a site-to-site connection. The TOE syncs its clock with the NTP server in order to provide proper timestamps.
SSH Terminal	A remote SSH client, allows an administrator to manage the TOE remotely on a secure management network.
Local Console	A local serial connection allows an administrator to manage the TOE locally in a secure physical environment.

Table 1 IT Environment Components

The TOE includes a Suricata engine. Suricata, an open-source threat detection engine, inspects and controls network traffic, examining both inbound and outbound data for potential security risks and breaches. Every data packet within the monitored network is scanned, decoded, and preprocessed, then assessed against defined access control and intrusion prevention criteria to identify and address unauthorized or harmful traffic, such as attempted attacks. If suspicious activity is detected, whether due to unexpected network behavior or a known threat signature, the system alerts a designated administrator and may also block the malicious traffic. The evaluation addressed the parts of the Suricata engine needed to meet the requirements including IPv4 and IPv6 events. The evaluation did not assess any predefined rules associated with the engine.

The TOE also provides firewall capabilities. The TOE filters traffic based upon administrator-defined rules and either permits or denies traffic accordingly and can log actions. The TOE supports many protocols for packet filtering including ICMPv4, ICMPv6, IPv4, IPv6, TCP and UDP.

The TOE was developed to meet its Protection Profile claims. Any functions outside those specified in this ST are not evaluated.

1.4.1.1 Physical Boundaries

The physical boundary of the TOE is the firmware as well as the hardware the firmware is installed on. The TOE is a single instance of virtual Berryville device running the Warden software v1.2 on a physical device running Ubuntu 22.04's KVM hypervisor. The physical hardware has an 11th Gen Intel(R) Core (TM) i7-1165G7 (Tiger Lake) CPU. All Ubunutu 22.04 functionality beyond the virtual device running on the KVM hypervisor is outside the scope of the evaluation. Ubuntu 22.04 should be configured that only the relevant services to the virtual device are open.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Warden:

- Security audit
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels
- Intrusion Prevention

1.4.1.2.1 Security audit

The TOE provides auditing capabilities to provide a secure and reliable record of all security relevant events, including administrative changes to the TOE. Any security relevant event is audited internally and then transmitted externally over a secure communication channel to an audit server via IPsec in real-time. All audited events have the necessary details like timestamp, event log, event code, and identity of the party involved to provide a comprehensive audit trail. Depending on the context of the audit, the identity may be the relevant user id, or remote IT entity involved in the event. All audits are protected from unauthorized deletion. The TOE's logrotate will rotate logs and once out of space, delete the oldest logs in order to make room for newer logs. The administrator may configure IPS auditing to limit the number of audits generated for similar events.

1.4.1.2.2 Cryptographic support

The TOE leverages OpenSSL library (version 3.0.10) executing on the TOE's Intel Core i7-1165G7 Processor library to provide cryptographic functions supporting secure administration access (via SSH), secure network traffic with VPN peers (via IKE/IPsec), and for secure communication to external systems such as audit log servers and NTP servers (also via IPsec). Functions include Key generation, key establishment, key distribution, key destruction, and cryptographic operations.

1.4.1.2.3 User data protection

The TOE has been designed to ensure that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, any additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

1.4.1.2.4 Firewall

The VM bridges the physical hardware's ethernet interface or additional USB interfaces to create virtual interfaces. Using these virtual interfaces, the TOE supports many protocols for packet filtering including icmpv4, icmpv6, ipv4, ipv6, tcp and udp. The firewall rules implement the SPD rules (permit, deny, bypass). Each rule can be configured to log status of packets pertaining to the rule. All codes under each protocol are implemented. The TOE supports FTP for stateful filtering.

Routed packets are forwarded to a TOE interface with the interface's MAC address as the layer-2 destination address. The TOE routes the packets using the presumed destination address in the IP header, in accordance with route tables maintained by the TOE.

IP packets are processed by the software, which associates them with application-level connections, using the IP packet header fields: source and destination IP address and port, as well as IP protocol. Fragmented packets are reassembled before they are processed.

The TOE mediates the information flows according to an administrator-defined policy. Some of the traffic may be either silently dropped or rejected (with notification to the presumed source).

The TOE's firewall and packet filtering capabilities are controlled by defining an ordered set of iptables rules. The rule specifies what communication will be allowed to pass and what will be blocked. It specifies the source and destination of the communication, what services can be used, and whether to log the connection.

1.4.1.2.5 Identification and authentication

The TOE maintains a single security administrator role. While the TOE allows unique users, each created user has the security administrator role. The TOE provides secure password-based for local administrators and password or public key based authentication for remote SSH administrators. The only functionality available to an administrator prior to authentication is viewing the warning banner. The TOE, supports passwords of varying lengths and allows an administrator to specify a minimum password length between 15 and 32 characters long. The password composition can contain all special characters as required by FIA_PMG_EXT.1.1.

Consecutive unsuccessful authentication attempts beyond a configurable limit will result in locking of the user for a specified duration of time.

The TOE provides secure connectivity between itself and a remote VPN peer, syslog server, or NTP server using IPsec with X.509 certificate-based authentication. X.509v3 certificates are stored internally and the store is protected by file permissions. X.509 certificates are manually loaded by the authorized administrator onto the TOE by an administrator. The TOE checks the revocations status of peer certificates via CRL. The TOE can generate certificate signing requests (CSRs) and accept the upload of the signed certificates.

1.4.1.2.6 Security management

The TOE maintains a single security administrator role that allows both local and remote administration for management of the TOE's security functions. TOE administrators manage the security functions of the TOE through a local console or SSH CLI. The security administrator is able to perform all management functions, including, but not limited to, modifying audit behavior, performing updates, managing crypto keys, managing firewall, IPS, and packet filtering rules.

1.4.1.2.7 Packet Filtering

The TOE provides packet filtering and secure IPsec tunneling. The TOE's netfilter kernel process bears responsibility for processing network packets and processes each packet against the netfilter rules governing each input and output chain. The tunnels can be established with
trusted VPN peers. More accurately, these tunnels are sets of security associations (SAs). The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used.

SAs are unidirectional and are established per the ESP security protocol. An authorized administrator can define the traffic that needs to be protected via IPsec by configuring access lists (permit, deny, log) and applying these access lists to interfaces using crypto map sets. The TOE processes incoming packets in netfilter's chain applying the administrator defined rulesets in order. See Section 1.4.1.2.4 above for more details.

1.4.1.2.8 Protection of the TSF

The TOE includes capabilities to protect itself from unwanted modification as well as protecting its persistent data.

The TOE does not store passwords in plaintext; they are obfuscated. The TOE does not support any command line capability to view any cryptographic keys generated or used by the TOE.

The TOE provides image integrity verification using a digital signature to validate the authenticity of the images before loading them. Upon every boot up, power on self-tests is conducted to validate the integrity of the software components as well as perform cryptographic known answer tests for the supported cryptographic algorithms. If power-up self-tests fail, the TOE halts boot. The TOE also allows administrator to manually configure of the TOE's clock or configuration of an NTP server, with which the TOE will synchronize its time. The TOE provides a timestamp for use with audit records, timing elements of cryptographic functions, and inactivity timeouts.

1.4.1.2.9 TOE access

The TOE offers a login banner which provides the administrator to ability to display a custom warning/access policy message as per the organization needs. The TOE provides the ability to configure an inactivity timeout which terminates the session beyond the inactivity period configured. An administrator can also terminate their own session.

1.4.1.2.10 Trusted path/channels

The TOE communicates to external components in a secure manner using IPsec for VPN peers, syslog servers, or NTP servers. The TOE also employs SSH to secure remote administrative sessions.

1.4.1.2.11 Intrusion Prevention

The TOE supports IPS functionality by leveraging the Suricata service. The TOE's intrusion detection and prevention system provides real-time monitoring and analysis of network traffic. The IPS will detect and respond to various types of network-based threats and attacks. IPS logs are generated and forwarded to Rsyslog. The IPS logs alert, drop, and reject actions to the syslog for traffic that matches a given rule. The TOE supports in-line and passive inspection modes using both anomaly and signature-based detection along with IP filtering based on blacklists & white-lists. Anomaly-based detection can be determined by thresholds. Signature-based detection can be determined by packet headers, string-based pattern-matching, attacks. and/or patterns. Due to the TOE leveraging open source service, Suricata, the TOE supports an extensive range of IPS detection events. Note that only the events explicitly defined in the SFRs have been evaluated,

1.4.2 TOE Documentation

Berryville Holdings, LLC Warden v1.2 CC Administrator Guide, 07/03/2025 (Admin Guide)

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - PP-Configuration for Network Device, Intrusion Prevention Systems (IPS), Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, version 1.2, August 18, 2023
    - Base PP:
        - collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 (NDcPP22e)
    - PP-Modules:
        - PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25 June 2020 (MOD_FW_1.4E)
        - PP-Module for Intrusion Prevention Systems (IPS), 1.0, 11 May 2021 (MOD_IPS_V1.0)
        - PP-Module for VPN Gateways, Version 1.3, 16 August 2023 (MOD_VPNGW_1.3)

Package	Technical Decision	Applied	Notes
CPP_ND_V2.2E	TD0800 - Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
CPP_ND_V2.2E	TD0792 - NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
CPP_ND_V2.2E	TD0790 - NIT Technical Decision: Clarification Required for testing IPv6	Yes	
CPP_ND_V2.2E	TD0738 - NIT Technical Decision for Link to Allowed-With List	Yes	
CPP_ND_V2.2E	TD0670 - NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	TLSC not claimed
CPP_ND_V2.2E	TD0639 - NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
CPP_ND_V2.2E	TD0638 - NIT Technical Decision for Key Pair Generation for Authentication	Yes	
CPP_ND_V2.2E	TD0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH	Yes	
CPP_ND_V2.2E	TD0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters	No	TLS not claimed
CPP_ND_V2.2E	TD0632 - NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
CPP_ND_V2.2E	TD0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	

CPP_ND_V2.2E	TD0592 - NIT Technical Decision for Local Storage of Audit Records	Yes	
CPP_ND_V2.2E	TD0591 - NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
CPP_ND_V2.2E	TD0581 - NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
CPP_ND_V2.2E	TD0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
CPP_ND_V2.2E	TD0572 - NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
CPP_ND_V2.2E	TD0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0570 - NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	DTLS not claimed
CPP_ND_V2.2E	TD0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
CPP_ND_V2.2E	TD0563 - NiT Technical Decision for Clarification of audit date information	Yes	
CPP_ND_V2.2E	TD0556 - NIT Technical Decision for RFC 5077 question	Yes	
CPP_ND_V2.2E	TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes	
CPP_ND_V2.2E	TD0547 - NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
CPP_ND_V2.2E	TD0546 - NIT Technical Decision for DTLS - clarification of Application Note 63	Yes	
CPP_ND_V2.2E	TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
CPP_ND_V2.2E	TD0536 - NIT Technical Decision for Update Verification Inconsistency	Yes	
CPP_ND_V2.2E	TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
CPP_ND_V2.2E	TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
MOD_CPP_FW_v1.4e	TD0827 - Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E	Yes	
MOD_CPP_FW_v1.4e	TD0551 - NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	Yes	
MOD_CPP_FW_v1.4e	TD0545 - NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	Yes	
MOD_IPS_V1.0	TD0902 - Updating RFC 2460 to 8200 in MOD_IPS_V1.0	Yes	
MOD_IPS_V1.0	TD0828 - Aligning MOD_IPS_V1.0 with CPP_ND_V3.0E	Yes	
MOD_IPS_V1.0	TD0722 - IPS_SBD_EXT.1.1 EA Correction	Yes	

MOD_IPS_V1.0	TD0595 - Administrative corrections to IPS PP-Module	Yes	
MOD_VPNGW_v1.3	TD0878: Updating FIPS 186-4 to 186-5 in MOD_VPNGW_V1.3	Yes	
MOD_VPNGW_v1.3	TD0838: PPK Configurability in FIA_PSK_EXT.1.1	No	SFR not claimed
MOD_VPNGW_v1.3	TD0824: Aligning MOD_VPNGW 1.3 with NDcPP 3.0E	Yes	
MOD_VPNGW_v1.3	TD0811: Correction to Referenced SFR in FIA_PSK_EXT.3 Test	No	SFR not claimed
MOD_VPNGW_v1.3	TD0781: Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3	No	SFR not claimed

Table 2 Technical Decisions

The following acronyms are used throughout the ST:

- CPP_ND_V2.2E - NDcPP22e
- MOD_CPP_FW_v1.4e -STFFW14e
- MOD_IPS_V1.0 - IPS10
- MOD_VPNGW_1.3- VPNGW13

2.1 Conformance Rationale

The ST conforms to the NDcPP22e/STFFW14e/IPS10/VPNGW13. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the NDcPP22e/STFFW14e/IPS10/VPNGW13 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP22e/STFFW14e/IPS10/VPNGW13 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP22e/STFFW14e/IPS10/VPNGW13 should be consulted if there is interest in that material.

In general, the NDcPP22e/STFFW14e/IPS10/VPNGW13 has defined Security Objectives appropriate for network devices and as such are applicable to the Berryville Warden TOE.

3.1 Security Objectives for the Operational Environment

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.

OE.COMPONENTS_RUNNING (applies to distributed TOEs only)
For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.

OE.CONNECTIONS TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic of monitored networks.

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 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.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the 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.TRUSTED_ADMIN TOE 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.VM_CONFIGURATION (applies to vNDs only)
For vNDs, the Security Administrator ensures that the VS and VMs are configured to
- reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and
- correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).
The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualisation features such as cloning, save/restore, suspend/resume, and live migration.
If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP22e/STFFW14e/IPS10/VPNGW13. The NDcPP22e/STFFW14e/IPS10/VPNGW13 defines the following extended requirements and since they are not redefined in this ST the NDcPP22e/STFFW14e/IPS10/VPNGW13 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1: IPsec Protocol - Per TD0800 and TD0824
- NDcPP22e:FCS_NTP_EXT.1: NTP Protocol
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
- STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering
- NDcPP22e:FIA_PMG_EXT.1: Password Management - per TD0792
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e/ VPNGW13:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW13:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW13:FPF_RUL_EXT.1: Packet Filtering Rules
- NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
- VPNGW13:FPT_TST_EXT.1: TSF Testing - per TD0824
- NDcPP22e/VPNGW13:FPT_TST_EXT.3: Self-Test with Defined Methods
- NDcPP22e/ VPNGW13:FPT_TUD_EXT.1: Trusted Update - per TD0824
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
- IPS10:IPS_ABD_EXT.1: Anomaly-Based IPS Functionality
- IPS10:IPS_IPB_EXT.1: IP Blocking
- IPS10:IPS_NTA_EXT.1: Network Traffic Analysis
- IPS10:IPS_SBD_EXT.1: Signature-Based IPS Functionality - per TD0722

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDcPP22e/STFFW14e/IPS10/VPNGW13. The refinements and operations already performed in the NDcPP22e/STFFW14e/IPS10/VPNGW13 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP22e/STFFW14e/IPS10/VPNGW13 and any residual operations have been completed herein. Of particular note, the NDcPP22e/STFFW14e/IPS10/VPNGW13 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP22e/STFFW14e/IPS10/VPNGW13. The NDcPP22e/STFFW14e/IPS10/VPNGW13 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Berryville Warden TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP22e/STFFW14e:FAU_GEN.1: Audit Data Generation
	IPS10:FAU_GEN.1/IPS: Audit Data Generation (IPS) - per TD0595
	VPNGW13:FAU_GEN.1/VPN: Audit Data Generation (VPN Gateway)
	NDcPP22e:FAU_GEN.2: User identity association
	NDcPP22e:FAU_STG.1: Protected audit trail storage
	IPS10:FAU_STG.1/IPS: Protected Audit Trail Storage (IPS Data)
	NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP22e:FCS_CKM.1: Cryptographic Key Generation
	VPNGW13:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication)
	NDcPP22e:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP22e:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP22e/VPNGW13:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP22e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP22e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP22e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP22e/VPNGW13:FCS_IPSEC_EXT.1: IPsec Protocol - Per TD0800
	NDcPP22e:FCS_NTP_EXT.1: NTP Protocol
	NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
FDP: User data protection	STFFW14e:FDP_RIP.2: Full Residual Information Protection
FFW: Firewall	STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering
FIA: Identification and authentication	NDcPP22e:FIA_AFL.1: Authentication Failure Management
	NDcPP22e:FIA_PMG_EXT.1: Password Management - per TD0792
	NDcPP22e:FIA_UAU.7: Protected Authentication Feedback

	NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP22e :FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
	VPNGW13:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP22e:FMT_MOF.1/Functions: Management of Security Functions Behaviour
	NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP22e:FMT_MOF.1/Services: Management of Security Functions Behaviour
	NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	VPNGW13:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP22e:FMT_SMF.1: Specification of Management Functions - per TD0631
	STFFW14e:FMT_SMF.1/FFW: Specification of Management Functions
	IPS10:FMT_SMF.1/IPS: Specification of Management Functions (IPS)
	VPNGW13:FMT_SMF.1/VPN: Specification of Management Functions
	NDcPP22e:FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	VPNGW13:FPF_RUL_EXT.1: Packet Filtering Rules
FPT: Protection of the TSF	NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
	VPNGW13:FPT_FLS.1/SelfTest: Failure with Preservation of Secure State (Self-Test Failures)
	NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
	NDcPP22e/VPNGW13:FPT_TST_EXT.1: TSF Testing - per TD0824
	VPNGW13:FPT_TST_EXT.3: Self-Test with Defined Methods
	NDcPP22e/ VPNGW13:FPT_TUD_EXT.1: Trusted update
FTA: TOE access	NDcPP22e:FTA_SSL.3: TSF-initiated Termination
	NDcPP22e:FTA_SSL.4: User-initiated Termination
	NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP22e:FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel - per TD0639
	VPNGW13:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications)
	NDcPP22e:FTP_TRP.1/Admin: Trusted Path - per TD0639
IPS: Intrusion Prevention	IPS10:IPS_ABD_EXT.1: Anomaly-Based IPS Functionality
	IPS10:IPS_IPB_EXT.1: IP Blocking
	IPS10:IPS_NTA_EXT.1: Network Traffic Analysis
	IPS10:IPS_SBD_EXT.1: Signature-Based IPS Functionality - per TD0722

Table 3 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP22e/STFFW14e:FAU_GEN.1)

NDcPP22e/STFFW14e: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).
- [***no other actions***];
d) Specifically defined auditable events listed in Table 2.

NDcPP22e/STFFW14e: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 2.

Requirement	Audit Event	Additional Contents
NDcPP22e:FAU_GEN.1		
NDcPP22e:FAU_GEN.2		
NDcPP22e:FAU_GEN_EXT.1		
NDcPP22e:FAU_STG.1		
NDcPP22e:FAU_STG_EXT.1		
NDcPP22e:FCS_CKM.1		
NDcPP22e:FCS_CKM.2		
NDcPP22e:FCS_CKM.4		
NDcPP22e:FCS_COP.1/DataEncryption		
NDcPP22e:FCS_COP.1/Hash		
NDcPP22e:FCS_COP.1/KeyedHash		
NDcPP22e:FCS_COP.1/SigGen		
NDcPP22e:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
NDcPP22e:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server
NDcPP22e:FCS_RBG_EXT.1		
NDcPP22e:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
STFFW14e:FDP_RIP.2		
STFFW14e: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

NDcPP22e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_PMG_EXT.1		
NDcPP22e:FIA_UAU.7		
NDcPP22e:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e: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
NDcPP22e:FIA_X509_EXT.2		
NDcPP22e:FIA_X509_EXT.3		
NDcPP22e:FMT_MOF.1/Functions		
NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	
NDcPP22e:FMT_MOF.1/Services		
NDcPP22e:FMT_MTD.1/CoreData		
NDcPP22e:FMT_MTD.1/CryptoKeys		
NDcPP22e:FMT_SMF.1	All management activities of TSF data.	
STFFW14e:FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	
NDcPP22e:FMT_SMR.2		
NDcPP22e:FPT_APW_EXT.1		
NDcPP22e:FPT_SKP_EXT.1		
NDcPP22e: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).
NDcPP22e:FPT_TST_EXT.1		
NDcPP22e:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	
NDcPP22e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	
NDcPP22e:FTA_SSL.4	The termination of an interactive session.	
NDcPP22e:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a	

	local session by the session locking mechanism.	
NDcPP22e:FTA_TAB.1		
NDcPP22e:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
NDcPP22e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	

Table 4 Auditable Events

5.1.1.2 Audit Data Generation (IPS) - per TD0595 (IPS10:FAU_GEN.1/IPS)

IPS10:FAU_GEN.1.1/IPS

The TSF shall be able to generate an IPS audit record of the following IPS auditable 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;
f) Totals of similar reactions occurring within a specified time period;
g) The events in the IPS Events table.
h) [***no other auditable events***].

IPS10:FAU_GEN.1.2/IPS

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, and;
b) For each IPS auditable event type, based on the auditable event definitions of the functional components included in the PP, information specified in column three of the IPS Events table.

Requirement	**Auditable Events**	**Additional Audit Record Contents**
FAU_GEN.1/IPS		
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. Aspect of the anomaly-based IPS policy rule that triggered the event (e.g. throughput, time of day, frequency, etc.). Network-based action by the TOE (e.g. allowed, blocked, sent reset to source IP, sent blocking notification to firewall).

IPS_IPB_EXT.1	Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.	Source and destination IP addresses (and, if applicable, indication of whether the source and/or destination address matched the list). TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset)
IPS_NTA_EXT.1	Modification of which IPS policies are active on a TOE interface. Enabling/disabling a TOE interface with IPS policies applied. Modification of which mode(s) is/are active on a TOE interface.	Identification of the TOE interface. The IPS policy and interface mode (if applicable).
IPS_SBD_EXT.1	Inspected traffic matches a signature-based IPS rule with logging enabled.	Name or identifier of the matched signature. Source and destination IP addresses. TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset).

Table 5 IPS Auditable Events

5.1.1.3 Audit Data Generation (VPN Gateway) (VPNGW13:FAU_GEN.1/VPN)

VPNGW13: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.

VPNGW13: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.

Requirement	Auditable Events	Additional Audit Record Content
VPNGW13:FAU_GEN.1/VPN		
VPNGW13:FCS_CKM.1/IKE		
VPNGW13:FCS_COP.1/DataEncryption		
VPNGW13:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
VPNGW13:FIA_X509_EXT.1/Rev		
VPNGW13:FIA_X509_EXT.2		
VPNGW13:FIA_X509_EXT.3		
VPNGW13:FMT_MTD.1/CryptoKeys		
VPNGW13:FMT_SMF.1/VPN	All administrative actions	

VPNGW13:FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport layer protocol
VPNGW13:FPT_FLS.1/SelfTest		
VPNGW13:FPT_TST_EXT.1		
VPNGW13:FPT_TST_EXT.3		
VPNGW13:FPT_TUD_EXT.1		
VPNGW13:FTP_ITC.1/VPN	Initiation of the trusted channel Termination of the trusted channel Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channel establishment attempt

Table 6 VPN Auditable Events

5.1.1.4 User identity association (NDcPP22e:FAU_GEN.2)

NDcPP22e:FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.5 Protected audit trail storage (NDcPP22e:FAU_STG.1)

NDcPP22e:FAU_STG.1.1
The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
NDcPP22e:FAU_STG.1.2
The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.1.1.6 Protected Audit Trail Storage (IPS Data) (IPS10:FAU_STG.1/IPS)

IPS10:FAU_STG.1/IPS.1
The TSF shall protect the stored IPS data from unauthorized deletion.
IPS10:FAU_STG.1/IPS.2
The TSF shall be able to prevent unauthorized modifications to the stored IPS data.

5.1.1.7 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1)

NDcPP22e: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.
NDcPP22e: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,***]
NDcPP22e:FAU_STG_EXT.1.3
The TSF shall [***overwrite previous audit records according to the following rule: [using logrotate to remove the oldest set of audit records in order to make space for newer records]***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

NDcPP22e:FCS_CKM.1.1
The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Appendix B.3,
- ECC schemes using 'NIST curves' [P-384] that meet the following: FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Appendix B.4,
- FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [RFC 3526]].

5.1.2.2 Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW13:FCS_CKM.1/IKE)

VPNGW13:FCS_CKM.1/IKE.1

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 B.4 for ECDSA schemes and implementing 'NIST curves' P-384 and [no other curves]***]
and
[***- no other key generation algorithm***]
and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.3 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

NDcPP22e: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' (TD0581 applied),
- 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] (TD0580 applied)].

5.1.2.4 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

NDcPP22e: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 [***single overwrite consisting of [zeroes]***];
- 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 [***logically addresses the storage location of the key and performs a [[seven]-pass] overwrite consisting of [zeroes]***]

that meets the following: No Standard.

5.1.2.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e /VPNGW13:FCS_COP.1/DataEncryption)

NDcPP22e/VPNGW13:FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] and [***no other***] mode and cryptographic key sizes [***256 bits***] and [***no other cryptographic key sizes***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***], and [***no other standards***].

5.1.2.6 Cryptographic Operation (Hash Algorithm) (NDcPP22e:FCS_COP.1/Hash)

NDcPP22e: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, SHA512***] and message digest sizes [160, ***256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP22e:FCS_COP.1/KeyedHash)

NDcPP22e:FCS_COP.1.1/KeyedHash

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

5.1.2.8 Cryptographic Operation (Signature Generation and Verification) (NDcPP22e:FCS_COP.1/SigGen)

NDcPP22e:FCS_COP.1.1/SigGen

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [assignment: 2048 bits or greater],

- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [384 bits]]

that meet the following:

[***-For RSA schemes: FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,***

- For ECDSA schemes: FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-384]; ISO/IEC 14888-3, Section 6.4].

5.1.2.9 IPsec Protocol - Per TD0800 (NDcPP22e/VPNGW13:FCS_IPSEC_EXT.1)

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.1

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

NDcPP22e/ VPNGW13: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.

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.3

The TSF shall implement [***tunnel mode***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-256 (RFC 3602), AES-GCM-256 (RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-384***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [***- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23], and [RFC 4868 for hash functions]***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-256 (specified in RFC 3602), AES-GCM-256 (specified in RFC 5282)***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [***- IKEv2 SA lifetimes can be configured by a Security Administrator based on [o length of time, where the time values can be configured within [1-24] hours]***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.8

The TSF shall ensure that [***- IKEv2 Child SA lifetimes can be configured by a Security***

Administrator based on [o length of time, where the time values can be configured within [1-8] hours]].

NDcPP22e/ VPNGW13: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 [**384 bits**] bits.

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv2***] exchanges of length [***- according to the security strength associated with the negotiated Diffie-Hellman group;***].

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s) [***[20 (384-bit Random ECP)]***].

NDcPP22e/ VPNGW13: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 [***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 [***IKEv2 CHILD_SA***] connection.

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1.13

The TSF shall ensure that [***IKEv2***] protocols perform peer authentication using [***ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***no other method***]. (TD0824 applied)

NDcPP22e/ VPNGW13: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]***].

5.1.2.10 NTP Protocol (NDcPP22e:FCS_NTP_EXT.1)

NDcPP22e:FCS_NTP_EXT.1.1

The TSF shall use only the following NTP version(s) [***NTP v4 (RFC 5905)***].

NDcPP22e:FCS_NTP_EXT.1.2

The TSF shall update its system time using [

- ***Authentication using [SHA1] as the message digest algorithm(s),***
- ***[IPsec] to provide trusted communication between itself and an NTP time source.***].

NDcPP22e:FCS_NTP_EXT.1.3

The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

NDcPP22e:FCS_NTP_EXT.1.4

The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.1.2.11 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

NDcPP22e:FCS_RBG_EXT.1.1

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

NDcPP22e:FCS_RBG_EXT.1.2

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

5.1.2.12 SSH Server Protocol - per TD0631 (NDcPP22e:FCS_SSHS_EXT.1)

NDcPP22e:FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol that complies with: RFC(s) 4251, 4252, 4253, 4254, [***4344, 5656, 6668***].

NDcPP22e:FCS_SSHS_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following user

authentication methods as described in RFC 4252: public key-based, [***password-based***]. (TD0631 applied)

NDcPP22e:FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**262130**] bytes in an SSH transport connection are dropped.

NDcPP22e:FCS_SSHS_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes256-gcm@openssh.com***].

NDcPP22e:FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [***rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp384***] as its public key algorithm(s) and rejects all other public key algorithms.

NDcPP22e:FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [***implicit***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

NDcPP22e:FCS_SSHS_EXT.1.7

The TSF shall ensure that [***ecdh-sha2-nistp384***] and [***diffie-hellman-group16-sha512***] are the only allowed key exchange methods used for the SSH protocol.

NDcPP22e:FCS_SSHS_EXT.1.8

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

5.1.3 User data protection (FDP)

5.1.3.1 Full Residual Information Protection (STFFW14e:FDP_RIP.2)

STFFW14e:FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.4 Firewall (FFW)

5.1.4.1 Stateful Traffic Filtering (STFFW14e:FFW_RUL_EXT.1)

STFFW14e:FFW_RUL_EXT.1.1

The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

STFFW14e:FFW_RUL_EXT.1.2

The TSF shall allow the definition of stateful traffic filtering rules using the following network protocol fields:

- ICMPv4
  - o Type
  - o Code
- ICMPv6
  - o Type
  - o Code
- IPv4
  - o Source address
  - o Destination Address
  - o Transport Layer Protocol
- IPv6 o Source address
  - o Destination Address
  - o Transport Layer Protocol
  - o [***no other field***]

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

and distinct interface.

STFFW14e: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.

STFFW14e:FFW_RUL_EXT.1.4

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

STFFW14e: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, [***no other protocols***] 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. [***no other protocols***].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [***session inactivity timeout***].

STFFW14e: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 [***counting***] packets which are invalid fragments;

b) The TSF shall drop and be capable of [***counting***] 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***].

STFFW14e: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.

STFFW14e:FFW_RUL_EXT.1.8
The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

STFFW14e:FFW_RUL_EXT.1.9
The TSF shall deny packet flow if a matching rule is not identified.

STFFW14e: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 [***counted, logged***].

5.1.5 Identification and authentication (FIA)

5.1.5.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

NDcPP22e:FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [**1 to 2147483647**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

NDcPP22e:FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

5.1.5.2 Password Management - per TD0792 (NDcPP22e:FIA_PMG_EXT.1)

NDcPP22e: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: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '[', ']', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '[', ']', '_', '`', '{', '|', '}', '~', '\', 'space'***];
b) Minimum password length shall be configurable to between [**15**] and [**32**] characters.

5.1.5.3 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

NDcPP22e:FIA_UAU.7.1
The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

5.1.5.4 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2)

NDcPP22e:FIA_UAU_EXT.2.1
The TSF shall provide a local [***password-based, SSH public key-based***] authentication mechanism to perform local administrative user authentication.

5.1.5.5 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

NDcPP22e: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;
- [***no other actions***].

NDcPP22e:FIA_UIA_EXT.1.2

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

5.1.5.6 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev)

NDcPP22e: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:

o 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.

o 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.

o 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.

o 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.

NDcPP22e: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.

5.1.5.7 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

NDcPP22e: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***].

NDcPP22e:FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall (choose one of): [***not accept the certificate***].

5.1.5.8 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3)

NDcPP22e:FIA_X509_EXT.3.1

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

NDcPP22e:FIA_X509_EXT.3.2

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

5.1.6 Security management (FMT)

5.1.6.1 Management of Security Functions Behaviour (NDcPP22e:FMT_MOF.1/Functions)

NDcPP22e:FMT_MOF.1.1/Functions

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

5.1.6.2 Management of security functions behaviour (NDcPP22e:FMT_MOF.1/ManualUpdate)

NDcPP22e:FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

5.1.6.3 Management of Security Functions Behaviour (NDcPP22e:FMT_MOF.1/Services)

NDcPP22e:FMT_MOF.1.1/Services

The TSF shall restrict the ability to start and stop services to Security Administrators.

5.1.6.4 Management of TSF Data (NDcPP22e:FMT_MTD.1/CoreData)

NDcPP22e:FMT_MTD.1.1/CoreData

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

5.1.6.5 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys)

NDcPP22e:FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.1.6.6 Management of TSF Data (VPNGW13:FMT_MTD.1/CryptoKeys)

VPNGW13:FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to manage the cryptographic keys and certificates used for VPN operation to Security Administrators.

5.1.6.7 Specification of Management Functions - per TD0631 (NDcPP22e:FMT_SMF.1)

NDcPP22e:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to start and stop services,***
- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),
-Ability to modify the behavior of the transmission of audit data to an external IT entity,
- Ability to manage the cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure thresholds for SSH rekeying,
- Ability to configure the lifetime for IPsec SAs,
- Ability to configure NTP,
- Ability to configure the reference identifier for the peer,
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,

-Ability to import X509v3 certificates to the TOE's trust store.
- Ability to manage the trusted public keys database].
(TD0631 applied)

5.1.6.8 Specification of Management Functions (STFFW14e:FMT_SMF.1/FFW)

STFFW14e:FMT_SMF.1.1/FFW
The TSF shall be capable of performing the following management functions: Ability to configure firewall rules.

5.1.6.9 Specification of Management Functions (IPS) (IPS10:FMT_SMF.1/IPS)

IPS10: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:
  - o Source IP addresses (host address and network address)
  - o Destination IP addresses (host address and network address)
  - o Source port (TCP and UDP) -- Destination port (TCP and UDP)
  - o Protocol (IPv4 and IPv6)
  - o 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 signaturebased IPS policies.

5.1.6.10 Specification of Management Functions (VPNGW13:FMT_SMF.1/VPN)

VPNGW13: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***]

5.1.6.11 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2)

NDcPP22e:FMT_SMR.2.1
The TSF shall maintain the roles: - Security Administrator.
NDcPP22e:FMT_SMR.2.2
The TSF shall be able to associate users with roles.
NDcPP22e:FMT_SMR.2.3
The TSF shall ensure that the conditions
- The Security Administrator role shall be able to administer the TOE locally;
- The Security Administrator role shall be able to administer the TOE remotely
are satisfied.

5.1.7 Packet Filtering (FPF)

5.1.7.1 Packet Filtering Rules (VPNGW13:FPF_RUL_EXT.1)

VPNGW13:FPF_RUL_EXT.1.1
The TSF shall perform Packet Filtering on network packets processed by the TOE.

VPNGW13: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)
  - o source address
  - o destination address
  - o protocol
- IPv6 (RFC 8200)
  - o source address
  - o destination address
  - o next Header (protocol)
- TCP (RFC 793)
  - o source port
  - o destination port
- UDP (RFC 768)
  - o source port
  - o destination port.

VPNGW13: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.

VPNGW13:FPF_RUL_EXT.1.4
The TSF shall allow the packet filtering rules to be assigned to each distinct network interface.

VPNGW13:FPF_RUL_EXT.1.5
The TSF shall process the applicable packet filtering rules (as determined in accordance with VPNGW13:FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

VPNGW13:FPF_RUL_EXT.1.6
The TSF shall drop traffic if a matching rule is not identified.

5.1.8 Protection of the TSF (FPT)

5.1.8.1 Protection of Administrator Passwords (NDcPP22e:FPT_APW_EXT.1)

NDcPP22e:FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

NDcPP22e:FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords.

5.1.8.2 Failure with Preservation of Secure State (Self-Test Failures) (VPNGW13:FPT_FLS.1/SelfTest)

VPNGW13:FPT_FLS.1.1/SelfTest
The TSF shall shut down when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

5.1.8.3 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1)

NDcPP22e:FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.8.4 Reliable Time Stamps - per TD0632 (NDcPP22e:FPT_STM_EXT.1)

NDcPP22e:FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.
NDcPP22e:FPT_STM_EXT.1.2
The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server, obtain time from the underlying virtualization system***].

5.1.8.5 TSF Testing - per TD0824 ((NDcPP22e/VPNGW13:FPT_TST_EXT.1)

NDcPP22e/VPNGW13: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 [***at no other time***]
- [***no other***]
to demonstrate the correct operation of the TSF: noise source health tests.
(TD0824 applied)
NDcPP22e/VPNGW13:FPT_TST_EXT.1.2
The TSF shall respond to [***all failures***] by [***/halting/***]. (per TD0824)

5.1.8.6 Self-Test with Defined Methods (VPNGW13:FPT_TST_EXT.3)

VPNGW13: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.
VPNGW13:FPT_TST_EXT.3.2
The TSF shall execute the self-testing through a TSF-provided cryptographic service specified in FCS_COP.1/SigGen.

5.1.8.7 Trusted Update - per TD0824 (NDcPP22e/VPNGW13:FPT_TUD_EXT.1)

NDcPP22e/ VPNGW13: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***].
NDcPP22e/ VPNGW13: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***].
NDcPP22e/ VPNGW13: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. (TD0824 applied)

5.1.9 TOE access (FTA)

5.1.9.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3)

NDcPP22e:FTA_SSL.3.1
The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.9.2 User-initiated Termination (NDcPP22e:FTA_SSL.4)

NDcPP22e:FTA_SSL.4.1
The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.1.9.3 TSF-initiated Session Locking (NDcPP22e:FTA_SSL_EXT.1)

NDcPP22e:FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.9.4 Default TOE Access Banners (NDcPP22e:FTA_TAB.1)

NDcPP22e: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.

5.1.10 Trusted path/channels (FTP)

5.1.10.1 Inter-TSF trusted channel - per TD0639 (NDcPP22e:FTP_ITC.1)

NDcPP22e:FTP_ITC.1.1

The TSF shall be capable of using [***IPsec***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[NTP server]***] 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.

NDcPP22e:FTP_ITC.1.2

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

NDcPP22e:FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [**syslog, NTP**].

5.1.10.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW13:FTP_ITC.1/VPN)

VPNGW13: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.

VPNGW13:FTP_ITC.1.2/VPN

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

VPNGW13:FTP_ITC.1.3/VPN

The TSF shall initiate communication via the trusted channel for (choose one of): [***remote VPN gateways or peers***].

5.1.10.3 Trusted Path - per TD0639 (NDcPP22e:FTP_TRP.1/Admin)

NDcPP22e: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.

NDcPP22e:FTP_TRP.1.2/Admin

The TSF shall permit remote Administrators to initiate communication via the trusted path.

NDcPP22e:FTP_TRP.1.3/Admin

The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

5.1.11 Intrusion Prevention (IPS)

5.1.11.1 Anomaly-Based IPS Functionality (IPS10:IPS_ABD_EXT.1)

IPS10:IPS_ABD_EXT.1.1
The TSF shall support the definition of [***anomaly ('unexpected') traffic patterns***] including the specification of
[***- thresholds***]
and the following network protocol fields:
[***all packet header and data elements defined in IPS_SBD_EXT.1***].

IPS10:IPS_ABD_EXT.1.2
The TSF shall support the definition of anomaly activity through [***manual configuration by administrators***].

IPS10:IPS_ABD_EXT.1.3
The TSF shall allow the following operations to be associated with anomalybased IPS policies:
- In any mode, for any sensor interface: [
  - ***allow the traffic flow,***
  - ***send a TCP reset to the source address of the offending traffic,***
  - ***send a TCP reset to the destination address of the offending traffic,***
  - ***send an ICMP [port] unreachable message,***]
- In inline mode:
  - allow the traffic flow
  - block/drop the traffic flow
  - and [***no other actions***].

5.1.11.2 IP Blocking (IPS10:IPS_IPB_EXT.1)

IPS10: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***].

IPS10: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]***].

5.1.11.3 Network Traffic Analysis (IPS10:IPS_NTA_EXT.1)

IPS10: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.

IPS10:IPS_NTA_EXT.1.2
The TSF shall process (be capable of inspecting) the following network traffic protocols:
- Internet Protocol version 4 (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.

IPS10: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: [**Ethernet**];
- Inline (data pass-through) mode: [**Ethernet**];
- Management mode: [**Ethernet**];

- [***no other interface types***].

5.1.11.4 Signature-Based IPS Functionality - per TD0722 (IPS10:IPS_SBD_EXT.1)

IPS10: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 [***no other field***].
- ICMP: type; code; header checksum; and [***[variable field based on type and code]***].
- ICMPv6: type; code; and header checksum.
- TCP: source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: source port; destination port; length; and UDP checksum].

IPS10: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;

IPS10:IPS_SBD_EXT.1.3

The TSF shall be able to detect the following header-based signatures (using fields identified in IPS10: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 DDoS Attack.

IPS10: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.

IPS10:IPS_SBD_EXT.1.5

The TSF shall allow the following operations to be associated with signaturebased IPS policies:
- In any mode, for any sensor interface: [
-- allow the traffic flow,
-- send a TCP reset to the source address of the offending traffic,
-- send a TCP reset to the destination address of the offending traffic,
-- send an ICMP [port] unreachable message]
- In inline mode:
o block/drop the traffic flow;
o and [***allow all traffic flow***].

IPS10: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

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic Functional Specification
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM Coverage
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 7 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE, including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e
The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d
The developer shall provide the TOE for testing.

AVA_VAN.1.1c
The TOE shall be suitable for testing.

AVA_VAN.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels
- Intrusion Prevention

6.1 Security audit

NDcPP22e/STFFW14e:FAU_GEN.1 & NDcPP22e:FAU_GEN.2:

The TOE is able to generate logs for a range of events as required by the requirements in Sections 5.1.1. The vendor's Administrative Guidance enumerates the TOE's audit records (generated by the TOE, as the TOE has only a single component and is not a distributed TOE). Each event log is unique with the date/time of the event, type of event, subject identity (e.g., IP address), and the outcome of the event. The TOE identifies cryptographic keys in audit records (records related to the generation/import, changing, or deleting of keys) by their issuer and subject Distinguished Names.

VPNGW13:FAU_GEN.1/VPN:
The TOE's VPN gateway functionality uses the same underlying audit mechanism (rsyslog) to generate its audits, and thus the TOE transmits VPN audits the same way as it does base-NDcPP audits.

IPS10:FAU_GEN.1/IPS:

The administrator can configure the TOE to log IPS data associated with the administrator enabled/configured polices.

The administrator may configure IPS auditing to limit the number of audits generated for similar events. This limiting can be configured via the following types:

- Threshold— this type sets a minimum count for a rule before it generates alerts
- Limit— this type prevents a flood of alerts by stopping alerts after a max count of generated alerts
- backoff— this type limits the alert output by using a backoff algorithm between generating alerts., meaning the frequency of generating audits is decreased the higher the count of a generated audit gets.

Signature based detection applies a set of pre-defined rules or custom rules. Rules are based on the Suricata rule format. Suricata rules are divided into two logical sections, the rule header and the rule options. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information, and a series of customizable rule options which cover all the fields described in IPS_SBD_EXT.1.

NDcPP22e:FAU_STG.1 & IPS10:FAU_STG.1/IPS:
The TOE stores its internal audit events in a log that is protected so that only the authorized administrator can read the audit events. The TOE has four gigabytes of internal storage for audit records. The audits are protected from unauthorized modification and deletion as only the authorized administrator can delete the audits. Modification of audits is not allowed.

NDcPP22e:FAU_STG_EXT.1:
The TOE is a standalone TOE that both stores its logs locally and transmits them remote to a syslog server over IPsec tunnel.

The TOE has four gigabytes of internal storage for audit records and should the TOE exhaust this storage space, it overwrites the oldest previous audit records with the new audit records. This is done via logrotate that will rotate logs and once out of space, delete the oldest logs in order to make room for newer logs.

The TOE transmits audit data via syslog transmitted within an IPsec tunnel. If a connection is established, then the audits are transported in near real-time.

6.2 Cryptographic support

The TOE's OpenSSL library (version 3.0.10) executing on the TOE's Intel Core i7-1165G7 Processor library possesses the following cryptographic algorithm certificates:

Requirements	Functions	Standard	CAVP Cert
Cryptographic key generation			
NDcPP22e:FCS_CKM.1	RSA schemes using 2048-bit keys	FIPS PUB 186-5	A6789
NDcPP22e:FCS_CKM.1	ECC schemes using 'NIST curves' P-384	FIPS PUB 186-5	A6789
NDcPP22e:FCS_CKM.1	FFC schemes using 'safe prime' group DH16 (4096)	NIST SP 800-56A	Tested with known good implementation
IKE Peer Auth Cryptographic key generation			
VPNGW13:FCS_CKM.1/IKE	ECC schemes using 'NIST curves' P-384	FIPS PUB 186-5	A6789
Cryptographic key establishment/distribution			
NDcPP22e:FCS_CKM.2	Elliptic curve-based key establishment schemes: P-384	NIST SP 800-56A	A6789
NDcPP22e:FCS_CKM.2	FFC schemes using 'safe prime' group DH16 (4096)	NIST SP 800-56A	Tested with known good implementation
Data Encryption			
NDcPP22e/VPNGW13:FCS_COP.1/DataEncryption	AES CBC, GCM (256 bits)	NIST SP 800-38A (CBC) NIST SP 800-38D (GCM)	A6789
Cryptographic hashing			
NDcPP22e:FCS_COP.1/Hash	SHA-1/256/384/512 (digest sizes 160, 256, 384 and 512 bits)	FIPS Pub 180-4	A6789
Keyed-hash message authentication			
NDcPP22e:FCS_COP.1/KeyedHash	HMAC-SHA-384 (key and output MAC size 384)	FIPS Pub 180-4	A6789
Cryptographic signature services			
NDcPP22e:FCS_COP.1/SigGen	RSA (RSA) Schemes using 2048 bits	FIPS PUB 186-5	A6789
NDcPP22e:FCS_COP.1/SigGen	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve P-384	FIPS PUB 186-5	A6789
Random bit generation			
NDcPP22e:FCS_RBG_EXT.1	CTR_DRBG (AES) with HW based noise source (256 bits)	NIST PUB 800-90A	A6789

Table 8 CAVP Certs

NDcPP22e:FCS_CKM.1 & VPNGW13:FCS_CKM.1/IKE:

The TOE provides key generation for asymmetric keys on all components and can generate ECDSA keys using NIST curve size P-384 and RSA keys of size 2048 [according to FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix B.4 and B.3 respectively]. The TOE supports DH group 16 key establishment scheme that meets standard RFC 3526, section 3 (for interoperability) as group 20 (ECP-384).

The does not introduce any TOE-specific extensions, processing or alternative implementations. However, the TOE does diverge from the sole "**should**" found in FIPS 186-5 section B.4.2 (within **Output:** step 2,). The TOE instead internally logs any error encountered during public/private key pair generation and does not output any *d* or *Q* values (in order to fail secure).

Purpose	SFR	Scheme	Size
IKE authentication	VPNGW13: FCS_IPSEC_EXT.1.13	ECDSA	P-384
IKE key exchange	VPNGW13: FCS_IPSEC_EXT.1.11	ECDH	P-384
SSH authentication	NDcPP22e:FCS_SSHS_EXT.1.5	RSA ECDSA	2048 P-384
SSH key exchange	NDcPP22e:FCS_SSHS_EXT.1.7	ECDH DH16	P-384 4096

Table 9 Key Usage Schemes

NDcPP22e:FCS_CKM.2:
Please see the table in NDcPP22e:FCS_CKM.1 above, which describes the asymmetric keys used for key establishment.

NDcPP22e:FCS_CKM.4:
The table below includes the cryptographic keys leveraged by the TOE and describes how and when the keys are zeroized.

Key	Storage (RAM/Flash)	Encrypted/ Plaintext	Destruction and when
IKE auth keys	RAM	Plaintext	Cleared w/ zeros after use
IKE auth keys	Flash	Plaintext	Stored persistently in the TOE's data partition, and cleared w/ a seven-pass overwrite (RNG and zeros) upon zeroization
IKE/ESP SA keys	RAM	Plaintext	Cleared w/ zeros after use
SSH host key	RAM	Plaintext	Cleared w/ zeros after use
SSH host key	Flash	Plaintext	Stored persistently in the TOE's data partition, and cleared w/ a seven-pass overwrite (RNG and zeros) upon zeroization
SSH session	RAM	Plaintext	Cleared w/ zeros after use

Table 10 Key Destruction

NDcPP22e/VPNGW13:FCS_COP.1/DataEncryption:
The TOE supports AES-256 in the GCM mode in SSH.

The TOE supports AES-256 CBC and GCM for its IPsec connections.

NDcPP22e:FCS_COP.1/Hash:
The TOE makes use of hashing for the integrity of IPsec/ESP traffic and for signature generation and verification (both SSH and IKE peer authentication and trusted updates). The TOE makes use of hashing for NTP authentication. The TOE supports cryptographic hashing services using SHA1, SHA-256, SHA-384, and SHA-512, with corresponding digest sizes of 160, 256, 384, and 512

NDcPP22e:FCS_COP.1/KeyedHash:
The TOE uses HMAC keys sized 384 bits with hashes SHA-384 with block size 1024 bits and with output MAC length 384 respectively as part of IKE/ESP integrity.

NDcPP22e:FCS_COP.1/SigGen:
The TOE generates ECDSA signatures (during SSH and IPsec/IKE peer authentication) using curves P-384. The TOE also verifies RSA 2048 signatures during SSH peer authentication and verifies ECDSA signature when verifying trusted updates using curve P-384.

NDcPP22e/ VPNGW13:FCS_IPSEC_EXT.1:

The following IPSEC TSS descriptions address the collective requirements of the base protection profile and those of the modules.

FCS_IPSEC_EXT.1.1:
The TOE implements the IPsec protocol as specified in RFC 4301, and the TOE matching incoming and outgoing traffic against packet filtering and SPD rules to determine whether traffic should bypass ESP encryption (BYPASS), have ESP encryption applied (PROTECT), or whether the traffic should be dropped (DISCARD). The administrator can configure these by specifying the syslog server to which the TOE connects.

The TOE routes all packets through the kernel's IPsec interface when a VPN is active. The kernel compares packets routed through this interface to the SPDs configured for the VPN to determine whether to PROTECT, BYPASS, or DISCARD each packet. The vendor designed the TOE to allow the administrator to configure SPDs by specifying the IP addresses of the syslog and VPN gateways/peers. Consequently, the TOE protects all traffic with VPN peers and TOE traffic to the syslog server.

FCS_IPSEC_EXT.1.3:
The TOE supports only tunnel mode. Any attempts by a peer to use transport mode will be rejected.

FCS_IPSEC_EXT.1.4:
The TOE implements both the AES-CBC-256 and AES-GCM-256 ciphers and implements the truncated HMAC-SHA-384 (which is truncated to 192 bits) algorithm for ESP integrity (the TOE uses HMAC-SHA-384 only when paired with the AES-CBC 256 cipher). The TOE's OpenSSL library implements the AES-CBC and HMAC-SHA-384 algorithms, while the TOE's Kernel Cryptography implements the AES-GCM algorithm.

FCS_IPSEC_EXT.1.5:
The TOE implements only IKEv2. Any attempts by a peer to use IKEv1 will be rejected. The TOE supports NAT traversal as specified in RFC 5996, section 2.23.

FCS_IPSEC_EXT.1.6:
The TOE uses AES-CBC-256 or AES-GCM-256 to encrypt its IKEv2 SAs.

FCS_IPSEC_EXT.1.7:
The TOE allows an administrator to provision the TOEs IKEv2 SA lifetime to a value between 1 and 24 hours.

FCS_IPSEC_EXT.1.8:
The TOE allows an administrator to provision the TOEs ESP SA lifetime to a value between 1 and 8 hours. Lifetime based on number of bytes is not supported.

FCS_IPSEC_EXT.1.9:
The TOE supports key exchange groups DH20 and generates a secret "x" of size 384 bits or greater bits.

FCS_IPSEC_EXT.1.10:
The TOE generates IKEv2 nonces using its DRBG and ensures a length of 384 bits.

FCS_IPSEC_EXT.1.11:
By default, the TOE supports only DH20 (ECP-384).

FCS_IPSEC_EXT.1.12:
The TOE only allows SA cipher strengths of 256 bits, hence the TOE's design inherently prevents a situation in where the ESP SA cipher strength exceeds that of the IKEv2 SA.

FCS_IPSEC_EXT.1.13:

The TOE supports ECDSA certificates for IKE peer authentication.

FCS_IPSEC_EXT.1.14:
The TOE supports Distinguished Name checking of VPN client certificates and checking of SAN:IPv4 (IPv4address in the Subject Alternative Name) for peer certificates.

NDcPP22e:FCS_NTP_EXT.1:

The TOE supports the syncing of its clock to a remote NTP server using NTP v4. The TOE communicates to the NTP server over an IPsec connection and additionally authenticates the NTP server using a SHA1 message digest algorithm. The TOE supports the configuration of at least 3 NTP servers and will not respond/sync to broadcast or multicast NTP messages.

NDcPP22e:FCS_RBG_EXT.1:
The TOE seeds its AES-256 CTR_DRBG using a 384-bit seed from a hardware entropy source.

NDcPP22e:FCS_SSHS_EXT.1:

The following SSH TSS descriptions address the collective requirements of the base protection profile.

NDcPP22e:FCS_SSHS_EXT.1.2:
The TOE supports public key authentication using the rsa-sha2-256, rsa-sha2-512 and ecdsa-sha2-nistp384 algorithms. The TOE matches an SSH client's presented public key with one within the TOE's authorized_keys file. The TOE also supports password-based authentication, and in the absence of a presented public key, the TOE prompts the client to supply a username and password.

NDcPP22e:FCS_SSHS_EXT.1.3:
The TOE inspects incoming SSH packets to check for those larger than 262,130 bytes in size and drops such packets.

NDcPP22e:FCS_SSHS_EXT.1.4:
The TOE supports the AES-256-GCM cipher mode.

NDcPP22e:FCS_SSHS_EXT.1.5:
The TOE's supports host keys using using the rsa-sha2-256, rsa-sha2-512 and ecdsa-sha2-nistp384 algorithms for authenticating to connecting SSH clients.

NDcPP22e:FCS_SSHS_EXT.1.6:
The TOE supports the "implicit" mode of integrity associated with AES-GCM, meaning AES-GCM already provides its own integrity and does not need and additional MAC.

NDcPP22e:FCS_SSHS_EXT.1.7:
The TOE supports groups DH16 (diffie-hellman-group16-sha512) and DH20 (ECP-384) ecdh-sha2-nistp384 as its key exchange algorithms.

NDcPP22e:FCS_SSHS_EXT.1.8:
The TOE supports configurable rekey limits of up to 1 hour and/or 1024 Megabytes and initiates a rekey when it encounters either threshold.

6.3 User data protection

STFFW14e:FDP_RIP.2:

The TOE has been designed to ensure that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, any additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

6.4 Firewall

FFW_RUL_EXT.1:

The VM bridges the physical hardware's ethernet interface or additional USB interfaces to create virtual interfaces. The TOE provides an information flow control mechanism using a rule base that comprises a set of security policy rules, i.e., the firewall security policy. The TOE enforces the firewall security policy on all traffic that passes through the engine, via its internal or external network Ethernet interfaces. The traffic is TCP, UDP, ICMPv4, ICMPv6, connections over IPv4 and IPv6. The TOE inspects and filters these protocols based upon their header fields as defined by their corresponding RFC (See Table 11).

The TOE only permits traffic to pass through that has been explicitly allowed by the firewall security policy, and implements packet defragmentation to enforce the policy on entire IP packets. Administrators using the Management Server define the firewall security policy rules.

Protocol	Related RFC[1]	Fields Inspected
ICMPv4	RFC 792	Type, Code
ICMPv6	RFC 4443	Type, Code
IPv4	RFC 791	Source Address, Destination Address, Transport layer protocol
IPv6	RFC 2460	Source Address, Destination Address, Transport layer protocol
TCP	RFC 793	Source Port, Destination Port
UDP	RFC 768	Source Port, Destination Port

Table 11 Protocols & Fields Filtered by the TOE

Any network traffic passed by the TOE must be explicitly allowed by a firewall rule or be part of an established session allowed by a rule, or it is dropped. This is true even in the case of attempts to flood a TOE interface (in which case some packets may be dropped, but no packets are ever passed that would violate policy).

All received network packets are processed by the TOE software module before transmission. The TOE software module does stateful filtering of the received network packets according to the configured traffic filtering rules. Protocol Agents are used for advanced processing of traffic that require special handling such as permitting an FTP data connection dynamically. The TOE software denies the traffic if the Protocol Agent cannot process the traffic. Incoming packets are dropped if a network packet cannot be processed due to insufficient memory. All incoming network packets are also discarded before the TOE software module has been loaded, and the TOE software module denies all traffic until the module has been configured. Network interfaces and routing are configured after the TOE software module has been loaded. If the configured firewall rules cannot be applied during startup, only the management network interface will be available and traffic through the firewall will be denied.

The TOE has been designed to ensure that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, any additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application). The TOE implements connection tracking to manage the information flow control decisions for connections (i.e., stateful sessions) rather than packets, providing increased performance and support for firewall features that require packet information above the IP level (e.g., TCP, UDP). The connection tracking mechanism stores the state information of each connection to allow packets belonging to an established connection to pass. The connection tracking uses the fields shown in the following table when determining whether a packet matches an allowed established session for the corresponding protocol. Connection tracking follows the standard TCP handshaking process (SYN, responding SYN-ACK, followed by ACK) to denote establishment of a stateful session, and the TOE's connection tracking will eliminate existing connections immediately, upon completion of the flow (in the case of TCP and FTP) or upon an inactivity timeout for the session. The TOE allows a max number of half-open TCP connections of 256 with a inactivity timeout of 3 minutes before being removed from connection tracking.

Protocol	Connection Tracking
TCP	Source & Destination Address, Source & Destination Port, Sequence Number, Flags

[1] Compliance with these RFCs is demonstrated by in-house compliance testing.

UDP	Source & destination address, source & destination port

Table 12 Connection Tracking Fields

Connection tracking works closely with the protocol agents to manage the information flow control decisions based on information attributes at the different networking layers through the application layer to decide whether a packet should be granted access or not.

The TOE follows a specific orderly algorithm to traverse the rule base for matching and filtering the traffic between its internal and external networks, first checking any open session with connection tracking and protocol agents, and then checking against any firewall rules if the packet is not part of an existing session. Any traffic that is not explicitly accepted by the security policy is rejected by the firewall. The structure of the rule base and the capabilities of its associated protocol agents enable the TSF to make the information flow control decisions.

Each rule comprises matching criteria and target actions. If the matching criteria is verified (i.e., a comparison matches) the TOE applies the target actions. Possible target actions include ACCEPT, DROP, REJECT, LOG and RETURN. Rules with the LOG target, can create a log each time they match. An administrator can specify that a rule applies to a specific interface by specifying the interface in the rule. Many more options are available to fine tune each rule's behavior..

The TOE compares the information attributes defined in Table 11 Protocols & Fields Filtered by the TOE with the matching criteria of the rule to determine whether to apply the rule. If applied, the target actions are implemented and the additional capabilities and flow control rules defined in **Error! Reference source not found.** are applied.

When in the evaluated configuration the TOE should block and log (or count in the case of invalid fragments) all of the following invalid traffic by default:

1. Packets which are invalid fragments, including a description of what constitutes an invalid fragment
2. Fragments that cannot be completely re-assembled
3. Packets where the source address is equal to the address of the network interface where the network packet was received
4. Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface
5. Packets where the source address is defined as being on a broadcast network
6. Packets where the source address is defined as being on a multicast network
7. Packets where the source address is defined as being a loopback address
8. Packets where the source or destination address of the network packet is a link-local address
9. Packets where the source or destination address of the network packet s defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4
10. Packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' as specified in RFC 3513 for IPv6
11. Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

The TOE is pre-configured with SYN attack prevention to help protect network hosts against SYN floods (half-open TCP connections), which are indicated by any traffic containing excessive incomplete connections to hosts on the network. SYN cookies are enabled by default and require no additional setup.

The default set of rules will begin dropping all SYN packets from a source IP that sends 100 in a second and continues dropping all SYN packets until under the threshold. When the dropping starts, one of the dropped packets will be logged each second.

Optional rules can be added to allow limiting the overall number of SYN packets for all users to no more than 200 per second, with an initial burst of 400 allowed. Anything over that rate will result in the excessive packets being dropped, and one packet logged per second.

The TOE allows a maximum number of half-open TCP connections of 256 with an inactivity timeout of 3 minutes before being removed from connection tracking. New SYN packets are dropped once the max is reached.

6.5 Identification and authentication

NDcPP22e:FIA_AFL.1:
The TOE allows the administrator to configure an admin lock out time between **1 to 2147483647 seconds.** The TOE tracks failed password authentication attempts for each user and will lock out the user for the administrator configured time, or until another administrator unlocks the user.

The TOE ensures that while failed logins results in lockout of remote users (administrators), the TOE does not lock out local administrators using host VM console interface after failed logins.

NDcPP22e:FIA_PMG_EXT.1:
The TOE supports passwords with a minimum and maximum length of between 15 and 32 characters consisting of the 95 ASCII printable characters (i.e., upper and lower case letters, numbers and the full set of special characters: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '"', '”', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '[', ']', '_', '`', '{', '|', '}', '~', '\', 'space'***]).

NDcPP22e:FIA_UAU.7:
The TOE obscures feedback to the local administrative user while authenticating.

NDcPP22e:FIA_UIA_EXT.1 & NDcPP22e:FIA_UAU_EXT.2:

The TOE provides both username/password as well as pubkey authentication for administrators (who connect via SSH). The TOE allows no actions prior to successful authentication (and successful authentication consists of either a valid public key authentication or a correct username/password combination) other than the displaying of the warning banner. The access through the TOE's Ethernet port constitutes remote administration and access through the host VM console constitutes local administration.

NDcPP22e:FIA_X509_EXT.1/Rev:
The TOE supports X.509v3 certificates for IPsec authentication. X.509v3 certificates are stored internally and the store is protected by file permissions. X.509 certificates are manually loaded by the authorized administrator onto the TOE by an administrator.

The authorized administrator configures the VPN peers for administrator and VPN communications, and specifies the DN or SAN with the peers cert. When an incoming request comes in, the TOE matches the peer's presented identifier to its configuration, to find the correct rule and then match the configured identifier to the peer certificate. The TOE then validates that it can construct a certificate path from the client's certificate through any intermediary CAs to the CA certificate specified by the user in the VPN configuration. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the certificates (e.g., checking its validity dates and that the CA flag is present in the basic constraints section for all CA certs). There are no exceptions to the rules for extendedKeyUsage fields. Assuming the certificates are valid, the TOE finally checks the revocation status of all using CRL. The TOE also ensures that explicit curve parameters are not used by the ECDSA certs. This is checked during IPsec authentication and when attempting to upload a certificate to the trust store.

NDcPP22e:FIA_X509_EXT.2:

The administrator can configure the certificate used during IKE authentication as part of the IPsec connection setup. If the TOE cannot establish a connected with a revocation server to check the status of a certificate in question, the TOE will not accept the certificate as valid.

NDcPP22e:FIA_X509_EXT.3:
The TOE can generate certificate signing requests (CSRs). The TOE includes a Common Name (CN), Organization, and Country in its generated CSR. Upon uploading the signed certificate back on to the TOE, the TOE verifies that the certificate was signed by a CA in its truststore.

6.6 Security management

NDcPP22e:FMT_MOF.1/Functions:

An authorized administrator can configure the external syslog server as well as view the configuration.

NDcPP22e:FMT_MOF.1/ManualUpdate:
The TOE allows only authorized administrators to initiate a manual update of the TOE's firmware.

NDcPP22e:FMT_MOF.1/Services:

The TOE restricts the ability to perform administrative functions such as starting and stopping services (IPsec tunnels, etc.) and configuration of the log export functions to the Security Administrator.

NDcPP22e:FMT_MTD.1/CoreData:
Only the administrator can configure TSF-related functions.

NDcPP22e:FMT_MTD.1/CryptoKeys & VPNGW13:FMT_MTD.1/CryptoKeys:

The TOE allows only security administrators to manage (i.e., import or delete) root CAs (used during IKE certificate authentication) and the certificates the TOE uses to authenticate itself to IKE peers. Only the security administrator can manage the SSH host keys and authorized_keys file for client public key authentication.

NDcPP22e:FMT_SMF.1 & STFFW14e:FMT_SMF.1/FFW & IPS10:FMT_SMF.1/IPS & VPNGW13:FMT_SMF.1/VPN:

Once authenticated, authorized administrators have access to the following security functions:

- Ability to administer the TOE locally and remotely;

- Ability to configure the access banner;

- Ability to configure the session inactivity time before session termination or locking;

- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;

- Ability to configure the authentication failure parameters for FIA_AFL.1;

- Ability to start and stop services,

- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),

- Ability to modify the behavior of the transmission of audit data to an external IT entity,

- Ability to manage the cryptographic keys,

- Ability to configure the cryptographic functionality,

- Ability to configure thresholds for SSH rekeying,

- Ability to configure the lifetime for IPsec SAs,

- Ability to configure NTP,

- Ability to configure the reference identifier for the peer,

- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,

- Ability to import X509v3 certificates to the TOE's trust store.

- Ability to manage the trusted public keys database

- Ability to configure firewall rules

- 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:

    o Source IP addresses (host address and network address)

    o Destination IP addresses (host address and network address)

o Source port (TCP and UDP) -- Destination port (TCP and UDP)

o Protocol (IPv4 and IPv6)

o 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.

- Definition of packet filtering rules

- Association of packet filtering rules to network interfaces

- Ordering of packet filtering rules by priority

NDcPP22e:FMT_SMR.2:
The TOE maintains a single security administrator role. All created accounts are security administrators. The security administrator can access the TOE both locally and remotely. Management of the TOE by the security administrator is identical for both local and remote, with one exception; account lockout due to exceeding the number of invalid password attempts does not apply to the local interface in order to prevent a denial-of-service.

6.7 Packet Filtering

VPNGW13:FPF_RUL_EXT.1:

VPNGW13:FPF_RUL_EXT.1.1:
The TOE's boot process brings up firewall rules (netfilter) prior to bringing up networking interfaces, and as such cannot send unfiltered traffic. The TOE's netfilter kernel process bears responsibility for processing network packets and processes each packet against the netfilter rules governing each input and output chain. Should netfilter fail, because it executes within the kernel, its failure would trigger a kernel panic and result in the TOE restarting.

VPNGW13:FPF_RUL_EXT.1.2 & VPNGW13:FPF_RUL_EXT.1.3 & VPNGW13:FPF_RUL_EXT.1.4:

The TOE implements a packet filtering policy implementation that can use the following fields in these RFC protocols:

The TSF shall allow the definition of Packet Filtering rules (permit, discard, and log) using the following network protocols and protocol fields:

- IPv4 (RFC 791)
  - o source address
  - o destination address
  - o protocol
- IPv6 (RFC 2460)
  - o source address
  - o destination address
  - o next Header (protocol)
- TCP (RFC 793)
  - o source port
  - o destination port
- UDP (RFC 768)

o source port
o destination port.

The VM bridges the physical hardware's ethernet interface or additional USB interfaces to create virtual interfaces. During development, the vendor determined compliance by examining the TOE's open-source implementation to ensure compliance and by also performing limited independent testing to ensure that the implementation could correctly filter based upon the above protocols and protocol fields.

VPNGW13:FPF_RUL_EXT.1.5:
The TOE processes incoming packets in netfilter's chain applying the administrator defined rulesets in order. The TOE also includes a set of default rules, which restrict the TOE to the minimum necessary traffic needed for the IPsec tunnel, SSH administration of the TOE, and the TOE's outgoing connections for syslog. The TOE also inspects packets to determine whether those packets are part of an established session, and if so, applies the administrator defined rulesets accordingly.

VPNGW13:FPF_RUL_EXT.1.6:
The TOE has a default discard rule which drops packets that match no existing or administrator defined rule, and the TOE's supports the full list of IPv4/IPv6 protocols and does not differ.

6.8 Protection of the TSF

NDcPP22e:FPT_APW_EXT.1:
The TOE stores administrative user passwords hashed with SHA-256. There exists no interface for an administrator to view passwords after they are configured.

NDcPP22e:FPT_SKP_EXT.1:
The TOE stores IKE authentication certificates and SSH public/private keys and while the TOE provides administrative access to update or replace these keys, the TOE provides no method to view or output these values (even to authorized administrators).

NDcPP22e:FPT_STM_EXT.1:
The TOE makes use of time when generating audit records (which include a timestamp) and when performing IKE certificate validation (checking certificate validity), and the TOE contains a Real-Time Clock (RTC). After the TOE's clock is set, the TOE can maintain accurate time using its internal clock. The clock can be set by an administrator, synced via an NTP server, or obtained from the underlying virtualization system.

VPNGW13:FPT_FLS.1/SelfTest & NDcPP22e/VPNGW13:FPT_TST_EXT.1 & VPNGW13:FPT_TST_EXT.3:

The TOE runs a set of self-test for both its OpenSSL library. The OpenSSL library covers its AES, SHA hashing, HMAC, ECDSA, RSA and DRBG algorithms. The self-test starts with known data (e.g., a known plaintext, key, and resulting ciphertext) and uses the data to ensure the algorithm works correctly. If any of these self-tests fail, the TOE halts its boot. In addition to these tests, the TOE also ensures the integrity of its firmware image. The TOE's bootloader verifies the ECDSA signature on its firmware image during boot. Finally, the TOE's noise source includes a health test to detect if the quality of the noise source degrades, and if so, halts outputting noise.

These tests together ensure that the TOE continues to operate correctly.

NDcPP22e/ VPNGW13:FPT_TUD_EXT.1:

The TOE provides an administrative command to check the installed firmware and further allows an administrator to updated the TOEs firmware by supplying a signed firmware update image. The TOE uses an existing, internal public key to verify the new image's signature (ensuring authenticity and integrity). If the digital signature is verified successfully, the TOE will continue to update automatically. If the digital signature verification fails, the update will not install, and the administrator should contact customer support. The TOE does not support automatic checking of updates.

6.9 TOE access

NDcPP22e:FTA_SSL.3 & NDcPP22e:FTA_SSL_EXT.1:
The TOE allows an administrator to configure a session inactivity time interval range of 0 and 2147483647 seconds for inactive administrator sessions. This covers both local and remote sessions on the TOE..

NDcPP22e:FTA_SSL.4:
The TOE allows administrators to terminate their remote or local sessions either through the logout command or by closing their SSH session.

NDcPP22e:FTA_TAB.1:
The TOE allows an administrator to set a banner that the TOE displays before each administrative session, which includes SSH and local console.

6.10 Trusted path/channels

NDcPP22e:FTP_ITC.1:FTP_ITC.1 & VPNGW13:FTP_ITC.1/VPN:
The TOE secures the connections to a remote syslog server and NTP server using IKEv2/IPsec. The TOE also uses IPsec to protect its communications with VPN clients. The TOE acts as an IKE/IPsec responder to VPN peers and an initiator when establishing a secure connection with a syslog server.

Cryptographic operations used in support of IPsec communication are described in Section 6.2.

NDcPP22e:FTP_TRP.1/Admin:
The TOE supports remote administrators through interactive SSH sessions

6.11 Intrusion Prevention

IPS10: IPS_IPB_EXT.1:

Suricata rules can be used to create IP whitelists and blacklists. TOE users can create a whitelist to explicitly allow traffic from only the listed known-good IP(s). A blacklist can also be created to block traffic from any defined known-bad IP (e.g., a known malicious IP). Whitelists and blacklists can be applied for single IP addresses as well as whole subnets. The pass and drop actions are leveraged for whitelists and blacklists. Only authorized Security Administrators can manage the whitelists and blacklists.

TOE users can view Suricata event logs outside of Rsyslog by displaying filtered alerts. Multiple filters can be applied to a single command to generate the desired output.

IPS10: IPS_ABD_EXT.1 & IPS10:IPS_NTA_EXT.1 & IPS10:IPS_SBD_EXT.1:

The TOE's intrusion detection and prevention system provides real-time monitoring and analysis of network traffic. The IPS will detect and respond to various types of network-based threats and attacks. IPS logs are generated and forwarded to Rsyslog. The IPS logs alert, drop, and reject actions to the syslog for traffic that matches a given rule. IPS logging is not configurable.

Suricata is capable of matching payloads across multiple packets without any additional configuration. TOE users can add and delete Suricata rules, as well as enable and disable available sources.

If desired, a dedicated management interface can be configured by excluding it from Suricata monitoring. Interface configuration can be completed using the following commands: interface internal/external suricata enable/disable.

The TOE contains only one set of interfaces, and as such, all Suricata rules apply to all interfaces where Suricata is enabled.

Suricata can operate in one of two modes, promiscuous (IDS), where traffic is passively monitored, and inline (IPS), where it can actively block/drop traffic.

A Suricata operating mode can be set using the following commands: suricata mode promiscuous/inline.

TOE users can manage Suricata rulesets and control which rules are used. Note the pre-defined rulesets were not evaluated for completeness/security so the administrator should defined their own rulesets. Note that some rulesets require subscriptions before they can be enabled.

Rulesets will be displayed with a unique name that combines the Vendor prefix with the rule name. Rulesets that require subscriptions will also contain a Subscription line with a URL leading to the vendor site.

In addition, the administrator can configure Denial-of-Service (DoS) rules to limit excess activity by users. The administrator then has the option to configure how such detected events are handled (pass, alert, or drop and log).

By default, Suricata comes with the et/open and tgreen/hunting rulesets enabled. The tgreen/hunting ruleset is a collection of Suricata IDS/IPS signatures developed by Travis Green, focusing on network anomaly detection and threat hunting. It is designed to help identify suspicious or unusual network activities that may indicate potential security threats. The et/open (Emerging Threats Open) ruleset includes a broad range of IDS/IPS rules designed to detect malware infections, botnet activity, exploit attempts, and command-and-control (C2) traffic. It covers many protocols such as HTTP, DNS, TLS, and FTP, and is regularly updated with signatures for known threats and suspicious network behaviors.

TOE users can add custom rules in Suricata. These rules contain a variety of actions, protocols, and options that can be tailored to create a unique environment. A rule/signature consists of the following:

- The action determines what happens when the rule matches.
- The header defines the protocol, IP addresses, ports, and direction of the rule.
- The rule options define the specifics of the rule.

The action determines what happens when the rule matches. Some valid actions are: alert, which generates an alert, pass, to stop further inspection of the packet, and drop, which drops the packet and generates an alert. The administrator can configure the traffic to pass, allow, or drop a packet and alert.

Valid actions are:

- alert - generate an alert.
- pass - stop further inspection of the packet.
- drop - drop packet and generate alert.
- reject - send RST/ICMP unreach error to the sender of the matching packet.
- rejectsrc - same as just reject.
- rejectdst - send RST/ICMP error packet to receiver of the matching packet.
- rejectboth - send RST/ICMP error packets to both sides of the conversation.

In "inline" mode, using any of the reject actions also enables drop.

Rules will be loaded in the order in which they appear in files, but will be processed in a different order. Signatures have different priorities, and the most important signatures will take precedence. The order is:

- pass
- drop
- reject
- alert

Within a specific action group, the priority setting can be used to further indicate the processing order. The rules with the lowest priority numbers will be processed first in ascending order.

The header defines the protocol, IP addresses, ports, and direction of the rule. Using these headers, the administrator can specify patterns that a payload within a packet must match to trigger the rule. The header protocol in a signature tells Suricata which protocol it is concerned with. Examples of basic protocols an administrator can choose between are: tcp (for tcp-traffic), udp, icmp, and ip (ip stands for 'all' or 'any'). These protocols apply to both IPv4 and IPv6.

There are a couple of additional TCP-related protocol options: tcp-pkt (for matching content in individual TCP packets) and tcp-stream (for matching content only in a reassembled TCP stream). As the TOE leverages, Suricata, a well-known and tested, open-source network intrusion detection and prevention system, conformance with the identified protocols has been ensured.

The rule options define the specifics of the rule. These are enclosed by parentheses and separated by semicolons. Some options have settings (such as msg), which are specified by the keyword of the option, followed by a colon, followed by the settings.

The TOE can perform string based detection by matching patterns in packets via the *content* rule.

The TOE can also be configured with rules to detect and mitigate a variety of popular network attacks, including:

- Fragmented IP packets
- Spoofed IP packets
- Fragmented ICMP packets
- Large ICMP packets
- Packets with improper TCP flags
- Malformed UDP packets
- Traffic floods of various protocols, such as TCP, UDP, and ICMP
- Port scans, host sweeps, and other types of reconnaissance traffic

Suricata threshold rules can limit how often a certain event triggers an alert. Thresholds are helpful by applying limits to how often an alert can be triggered when the same event occurs multiple times. The threshold settings can be used with any of the Suricata rules.

There are four threshold modes:

- threshold - sets a minimum threshold for a rule before it generates alerts.
- limit - prevents a flood of alerts by limiting the number of alerts.
- both - combines threshold and limit to control when alerts are generated.
- backoff - limits the alert output by using a backoff algorithm between alerts.

Also useful for anomaly detection, a threshold can monitor the network for unusual events or trends. It's particularly useful for preventing traffic flooding from and/or to a specific IP address.