Viasat Secure VPN v1.1.7 Security Target

Document Number: 1438289

Version: 2.6

December 13, 2023

Prepared For:

Viasat, Inc.
2426 Town Garden Road
Carlsbad, CA 92009

Prepared By:
Michael C. Baron
UL Verification Services Inc.

Table of Contents

1. Security Target (ST) Introduction

The structure of this document is defined by CC v3.1r5 Part 1 Annex A.2, “Mandatory contents of an ST”:

- Section 1 contains the ST Introduction, including the ST reference, Target of Evaluation (TOE) reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions

1.1 Security Target Reference

ST Title:	Viasat Secure VPN v1.1.7 Security Target
ST Version Number:	Version 2.6
ST Author(s):	Michael C. Baron
ST Publication Date:	December 13, 2023
Keywords:	Network Device, VPN Gateway, IPsec, vND, Virtual Network Device

1.2 Target of Evaluation Reference

TOE Developer:	Viasat, Inc. 6155 El Camino Real Carlsbad, CA 92009-1699
TOE Identification:	Viasat Secure VPN v1.1.7

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a VPN Gateway Virtual Network Device.

The TOE is fits under the definition of ‘Case 1’ Evaluated Configuration as described in [cPP] Section 1.2.

The TOE fits under the definition of ‘Use Case 1’ as described in [VPN] Section 1.4.

1.3.2 TOE Usage

Viasat’s Secure VPN virtual Network Device (TOE) is intended to provide bump-in-the-wire IPsec encryption to virtual or physical systems deployed behind the device on the Plaintext network.

The device supports Gateway (GW) to GW IPsec encryption. The sources and destinations that send data over the IPsec tunnel provided by the device and its remote peer are not aware of their existence.

The TOE is a Virtualized Network Device executing on Windows Hyper-V virtual machine manager running on the Windows 10 Pro 22H2 Operating System. The TOE is a VPN Gateway, providing an external (black network) interface facing the WAN.

Figure 1 below is a block diagram of the major security functionality and interfaces of the TOE. This figure depicts the following information:

- The TOE is demarked by the red dashed line
- Logical and physical interfaces are identified
- Components of the OE are identified
    - Local ("CLI") and Remote Management Interface ("RMI")
        - RMI is the Trusted Path TLS/HTTPS protected interface
    - Remote syslog server
    - Remote IPsec Gateway Peer
    - Data flows from Red Network to/from Black Network

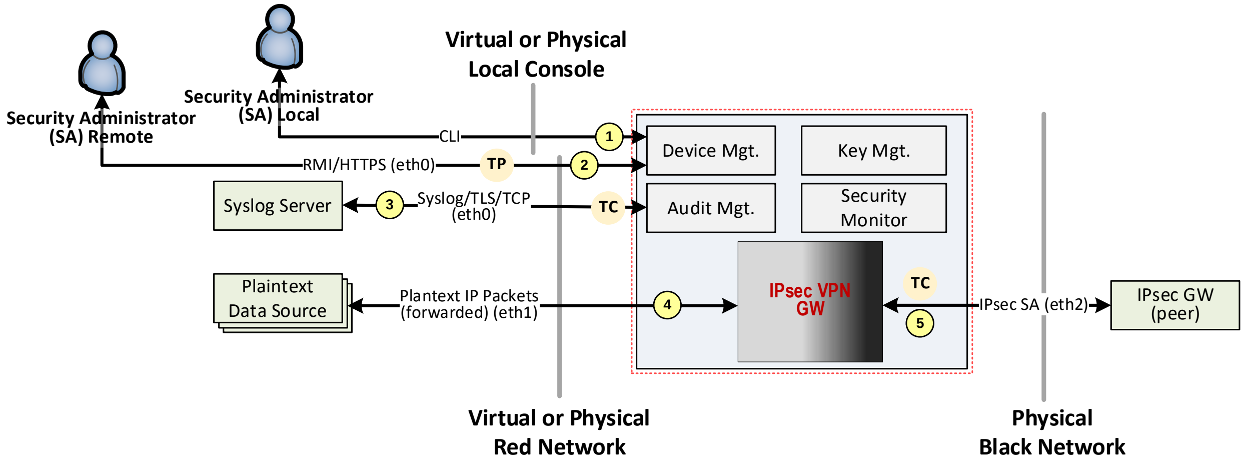

Figure 1 - TSF Block Diagram

The TOE is capable of importing cryptographic keys that are generated outside the boundary of the TOE via a .pkcs12 file. The TOE, however, does not provide functionality to validate the security strength of the imported private key, therefore, this functionality is unevaluated.

The TOE supports IPsec traffic protection of only IPv4 packets. IPv6 packets can be received by the device, but only 'Drop' and 'Bypass' traffic processing rules are supported for such packets.

1.3.3 TOE Major Security Features Summary

- Audit
- Communication
- Cryptography
- Identification and Authentication
- Security Management
- Protection of the TSF
- Packet Filtering
- TOE Access

- Trusted Path/Channels

1.3.4 TOE IT environment hardware/software/firmware requirements

The TOE requires the following hardware and software platform to operate:

- Hardware Platform
  - Dell XPS 8940 Server Hardware
    - Processor: 11th Gen Intel Core i5-1140 @ 2.6Ghz
    - RAM: 16Gb Memory
    - Hard drive: 1TB mechanical (spinning) SATA drive
- Software Platform
    - Windows 10 Pro 22H2
    - Microsoft (MS) Hyper-V Type 1 hypervisor Virtual Machine Manager (VMM)
      - Note that Hyper-V is bundled with the Windows Operating System

The TOE requires the following components to be present in the OE to support the TSF:

- Syslog server conformant to RFCs 5424 (Syslog over TCP)
- CRL Distribution Point capable of transport of CRLs over HTTP

The TOE requires the following software to be present in the OE for remote management of the TSF:

- "REST API client tool":
  - HTTPS client must support establishing an HTTPS session using TLSv1.2 with the following ciphersuite:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - This tool must have the ability to submit customized HTTP requests to the TSF's RMI API (please see the 'RMI REST API REFERENCE' in Appendix A of the guidance documentation for specific details on accessing this API).
  - The abilities of the HTTP client should include, but not be limited to:
    - GET, PUT, DELETE and POST requests to specific HTTP URL endpoints
    - JSON content format in the body of the request
    - Customization of Request, General and Representation headers
  - A API platform tool such as https://www.postman.com/ would suffice

The TOE requires the following components to be present in the OE for local management of the TSF:

  - USB Keyboard and monitor (HDMI or DisplayPort) to interface with the TOE platform hardware

1.4 Target of Evaluation Description

1.4.1 TOE Physical Scope

The TOE consists of the following software:

- Viasat's Secure VPN vND version 1.1.7
  - Delivery process:

  - Available for download from the Viasat customer web portal which requires username/password credentials of validated customers of the TOE. The format of the download is '.vhdx' file.

The guidance documentation that is part of the TOE is listed in Section 9 "References" within Table 15: TOE Guidance Documentation. The guidance documentation is downloaded from the NIAP website in PDF form.

1.4.2 Evaluated Configuration

The TOE has only one evaluated configuration. The evaluated configuration consists of the Hardware Platform and Software Platform listed in Section 1.3.4 and the TOE software as described in Section 1.4.1, in conjunction with the administrative configuration as described in the Guidance Documentation. The evaluated configuration is a 'Case 1' evaluated configuration as described in [cPP] Section 1.2, where the TOE is represented by the vND alone. The evaluated configuration includes the vND and the Virtualization System (VS) where the VS encompasses the virtual hardware abstraction, the hypervisor or virtual machine manager (VMM) and the physical chassis.

No other evaluated configurations are expressed or implied.

1.4.3 TOE Logical Scope

The logical boundary of the TOE include those security functions implemented exclusively by the TOE. These security functions are listed in Section 1.3.3 above and are further described in the following subsections. A more detailed description of the implementation of these security functions are provided in Section 7 "TOE Summary Specification".

1.4.3.1 Audit

- The TOE will audit all events and information defined in **Table 3**: Auditable Events.
- The TOE will also include the identity of the user that caused the event (if applicable), date and time of the event, type of event, and the outcome of the event.
- The TOE protects storage of audit information from unauthorized deletion.
- The TOE prevents unauthorized modifications to the stored audit records.
- The TOE can transmit audit data to/receive data from an external IT entity using the TLS protocol.
- The TOE performs audit log rotation when the local storage of audit records is full.
- The TOE counts the number of audit records that are overwritten when the local storage space for audit records is full.

1.4.3.2 Cryptographic Operations

The TSF performs the following cryptographic operations:

For TLS as a client and server, supporting the following cryptographic algorithms:

- Supports the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ciphersuite consisting of the following cryptographic services:
  - ECDSA digital signature generation/verification
  - AES-256 in GCM mode for bulk data ciphering
  - ECDHE key exchange utilizing the secp384r1 elliptic curve
  - SHA-384 hashing primitive
  - HMAC-SHA2-384 for keyed hashing

The TLS client TSF supports mutual authentication utilizing x.509v3 PKI.

For IPsec, the TSF supports the following:

- ECDSA digital signature generation/verification for IKEv2 supporting NIST P-256 and P-384 curves.
- AES-GCM-256 algorithm for encryption and message authentication for the IPsec ESP protocol
- AES-GCM-256 or AES-CBC-256 to protect the IKEv2 payload
- HMAC-SHA-384 to authenticate the IKEv2 payload
- Diffie-Hellman groups 19 and 20 for use in IKEv2
- IPv4 only; IPv6 is not supported for IPsec

The TSF utilizes a CTR_DRBG using AES-256, as its source for secure random bit generation.

The Trusted Update TSF utilizes ECDSA digital signatures associated with x.509v3 certificates using P-384 curve.

The TSF zeroizes all plaintext secret and private cryptographic keys and CSPs once they are no longer required.

1.4.3.3 Identification and Authentication

- The TSF supports passwords consisting of alphanumeric and special characters.
- The TSF also allows the Security Administrator (SA) to set a minimum password length.
- The TSF will lock out offending accounts that fail to successfully authenticate after an administratively defined number failed authentication attempts that the remote management interface. The offending account will be unlocked after an administratively configurable amount of time elapses.
- The TSF provides a local console management interface that is accessible via username and password authentication
- The TSF does not echo back characters input for the password at the local console
- The TSF utilizes x.509v3 certificates to identify itself to remote management users via the trusted path (HTTPS Server).
- The TSF utilizes x.509v3 certificate-based authentication to support a mutually authenticated trusted channel to a remote audit logging server (TLS client with mutual authentication).
- The TSF utilizes x.509v3 certificates for authentication of system software updates.
- The TSF supports the generation of Certificate Signing Requests.
- The TSF requires all administrative-users to authenticate before allowing the user to perform any actions other than:
  - Viewing the warning banner
  - Automated generation of cryptographic keys
  - ICMP echo reply (when configured in packet filtering table by the SA)
  - Responding to ARP requests with ARP replies
  - packet forwarding through the IPsec tunnel (when configured by the SA)
  - packet forwarding through BYPASS packet filtering table (when configured by the SA)

1.4.3.4 Security Management

The TSF stores and protects the following data:

- Local audit records, user account data, and local authentication data (such as administrator passwords)
- Cryptographic keys including symmetric keys, and private keys.

There is one class of user on the TOE:

- Security Admin user

Management of the TSF:

- The administrator can perform manual updates, determine the behavior of or modify the behavior of the handling of audit data, modify the behavior of the TSF, enable or disable services offered by the TOE, manage TSF data, modify, delete, generate or import cryptographic keys, configure the access banner, manage packet filtering and configure the session inactivity timeout period.
- The administrator may perform these functions locally or remotely via the CLI or RMI

1.4.3.5 Protection of the TSF

- The TSF prevents the reading of secret and private keys.
- The TOE provides reliable time stamps for itself.
- The TOE runs a suite of self-tests during the initial start-up (upon power on) to demonstrate the correction operation of the TSF.
- The TOE provides a means to verify firmware/software updates to the TOE using a digital signature mechanism prior to installing those updates.

1.4.3.6 Packet Filtering

- The TSF can be configured to filter network packets based on IPv4, IPv6, TCP and UDP protocols.
    - The TOE can only DROP and LOG IPv6 packets.
- The TSF can be configured to log network packets that match a packet filter rule
- The TSF processes packet filter rules in an administratively defined order
- Packet filtering can be applied to any network interface of the TOE
- The TSF has a final ‘drop’ rule if no rule matches the packet being processed

1.4.3.7 TOE Access

- The TOE, for local interactive sessions, terminates active session after an Authorized Administrator-specified period of session inactivity.
- The TOE terminates a remote interactive session after an Authorized Administrator-configurable period of session inactivity.
- The TOE allows Administrator-initiated termination of the Administrator’s own interactive session.
- Before establishing an administrative user session, the TOE is capable of displaying an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

1.4.3.8 Trusted Path/Channels

- The TOE uses IPsec, and TLS to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.
- The TOE permits the TSF, or the authorized IT entities to initiate communication via the trusted channel.
- The TOE permits remote administrators to initiate communication via the trusted path.

- The TOE requires the use of the trusted path for initial administrator authentication and all remote administration actions.

1.5 Notation, formatting, and conventions

The notation, formatting, and conventions used in this Security Target are defined below; these styles and clarifying information conventions were developed to aid the reader.

Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification.

The notation conventions that refer to iterations, assignments, selections, and refinements made in this Security Target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a Protection Profile.

The notation used in those PP to indicate assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document. Additionally, obvious errors in the PP are corrected and noted as such.

The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; Section 8.1, “Operations” as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

Iterations performed by the ST author are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1). Iterations performed by the PP author are indicated by a slash followed by a short description, e.g. FCS_COP.1/Hash.

Assignments are identified with **bold text.**

Selections are identified with underlined text**.** Selections made within selections (so called ‘nested selections’) are identified with a double underline.

Refinements that add text use ***bold and italicized text*** to identify the added text***.*** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

2. Conformance Claims

2.1 Common Criteria Conformance

This Security Target and TOE are conformant to the Common Criteria Version 3.1 Release 5, CC Part 2 extended [C2], and CC Part 3 conformant [C3].

2.2 Protection Profile Configuration Conformance

This Security Target claims exact conformance to the following PP-Configuration:

- PP-Configuration for Network Device and Virtual Private Network (VPN) Gateways, v1.2
    - This PP configuration includes the following components:
        - Base-PP: collaborative Protection Profile for Network Devices, Version 2.2e (CPP_ND_V2.2E) [cPP].
            - Note: This Protection Profile will be referred to as cPP or PP for convenience throughout this Security Target.
        - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways Version 1.2 (mod_vpngw_v1.2) [VPNGW].
            - Note: This PP-Module will be referred to as VPNGW for convenience throughout this Security Target.

Error! Reference source not found. below lists the Technical Decisions published for [cPP] and [VPNGW], and includes an indication of their applicability to the TOE:

Table 1: Technical Decisions

TD	TD Title	Applies to TOE?
0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes
0790	NIT Technical Decision: Clarification Required for testing IPv6	Yes
0771	Correction to FIA_PSK_EXT.3 EA	No
0738	NIT Technical Decision for Link to Allowed-With List	Informational
0723	Correction to ECDSA Curve Selection	Yes
0683	RFC 2460 to be replaced with RFC 8200	Yes
0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Yes
0657	IPSEC_EXT.1.6 GCM support for VPN GW	Yes
0656	Missing EAs for VPN GW Optional Headend SFRs	No
0639	NIT Technical Decision for Clarification for NTP MAC Keys	No
0638	NIT Technical Decision for Key Pair Generation for Authentication	Yes
0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No
0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	Yes
0633	NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes
0632	NIT Technical Decision for Consistency with Time Data for vNDs	Yes
0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	No
0592	NIT Technical Decision for Local Storage of Audit Records	Applies to all TOEs.

Table 1: Technical Decisions

TD	TD Title	Applies to TOE?
0591	NIT Technical Decision for Virtual TOEs and hypervisors	General PP update that applies to Virtualized TOEs + Acronym update. Essentially applies to all evaluations.
0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes
0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	No
0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Informational
0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Informational; applies to all TOEs
0570	NiT Technical Decision for Clarification about FIA_AFL.1	Informational; applies to all TOEs
0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.	Yes
0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	Applies to all TOEs
0563	NiT Technical Decision for Clarification of audit date information	Applies to all TOEs
0556	NIT Technical Decision for RFC 5077 question	Yes
0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	Informational
0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes
0546	NIT Technical Decision for DTLS - clarification of Application Note 63	Applies to all TOEs
0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes
0536	NIT Technical Decision for Update Verification Inconsistency	Yes
0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No
0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes

2.3 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements or security assurance requirements packages, neither as package-conformant or package-augmented.

2.4 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
    - All threats defined in the cPP and VPNGW module;
    - No additional threats have been defined in this ST.
- Organizational Security Policies
    - All OSP defined in the cPP and VPNGW module are carried forward to this ST;
    - No additional OSPs have been defined in this ST.
- Assumptions
    - All assumptions defined in the cPP and VPNGW module for a standalone TOE are carried forward to this ST;
    - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
    - All objectives defined in the cPP and VPNGW module for a standalone TOE are carried forward to this ST. Optional and selection based SFRs defined in the cPP are carried forward to this Security Target as required by the cPP.
- All mandatory SFRs and SARs defined in the cPP and VPNGW module are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the cPP and VPNGW module have been properly instantiated in this Security Target; therefore, this ST demonstrates exact compliance to the cPP and VPNGW module.

3. Security Problem Definition

3.1 Threats

The following section defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP unchanged.

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.

T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.

T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.

T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices.

T.SECURITY_FUNCTIONALITY_FAILURE

An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

T.DATA_INTEGRITY

Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained within the communications may be susceptible to a loss of integrity.

T.NETWORK_ACCESS

Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network.

From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network.

From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled email servers, or, that access to the mail server must be done over an encrypted link.

T.NETWORK_DISCLOSURE

Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information.

From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information.

From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.

T.NETWORK_MISUSE

Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network.

From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network.

Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services.

From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.

T.REPLAY_ATTACK

If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a “replay” attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions:

- Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome
- No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these

3.2 Organizational Security Policies

The following section defines the organizational security policies which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP unchanged.

P.ACCESS_BANNER

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP unchanged.

A.PHYSICAL_PROTECTION

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

A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.

A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE’s trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).

A.REGULAR_UPDATES

The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE

The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.

A.RESIDUAL_INFORMATION

The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

A.VS_TRUSTED_ADMINISTRATOR

The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.

A.VS_REGULAR_UPDATES

The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.VS_ISOLATON

For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.

A.VS_CORRECT_CONFIGURATION

For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.

A.CONNECTIONS

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

4. Security Objectives

4.1 Security Objectives for the TOE

O.ADDRESS_FILTERING

To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement packet filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) or receiving (destination) applicable network traffic as well as on established connection information.

O.AUTHENTICATION

To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.

O.CRYPTOGRAPHIC_FUNCTIONS

To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.

O.FAIL_SECURE

There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.

O.PORT_FILTERING

To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.

O.SYSTEM_MONITORING

To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).

O.TOE_ADMINISTRATION

TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.

4.2 Security Objectives for the Operational Environment

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.

OE.NO_THRU_TRAFFIC_PROTECTION

The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.

OE.UPDATES

The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

OE.ADMIN_CREDENTIALS_SECURE

The Administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

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.VM_CONFIGURATION

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.

OE.CONNECTIONS

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

5. Extended Components Definition

As stated in Section 2, this Security Target claims exact conformance to the referenced cPP and modules. As such, the extended components definition is contained in the cPP and modules claimed. In this case the cPP is Part 3 conformant and so there are no extended SARs defined.

6. Security Requirements

This section describes the security functional and assurance requirements for the TOE.

6.1 Security Function Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the cPP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 2: Security Functional Requirements

SFR	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.1/VPN	Audit Data Generation (VPN Gateway)
FAU_GEN.2	User Identity Association
FAU_STG.1	Protected audit trail storage
FAU_STG_EXT.1	Protected Audit EventStorage
FAU_STG_EXT.2 /LocSpace	Counting lost audit data
FCS_CKM.1	Cryptographic Key Generation (Refinement)
FCS_CKM.2	Cryptographic Key Establishment (Refinement)
FCS_CKM.1/IKE	Cryptographic Key Generation (for IKE Peer Authentication)
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_IPSEC_EXT.1	IPsec Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Protocol with Authentication
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_AFL.1	Authentication Failure Management (Refinement)
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation

Table 2: Security Functional Requirements

SFR	Description
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1 /ManualUpdate	Management of security functions behavior
FMT_MOF.1 /Services	Management of security functions behavior
FMT_MTD.1 /CoreData	Management of TSF Data
FMT_MTD.1 /CryptoKeys	Management of TSF data
FMT_SMF.1	Specification of Management Functions
FMT_SMF.1/VPN	Specification of Management Functions (VPN Gateway)
FMT_SMR.2	Restrictions on security roles
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_FLS.1/SelfTest	Fail Secure (Self-Test Failures)
FPF_RUL_EXT.1	Rules for Packet Filtering
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TST_EXT.1	TSF Testing (Extended)
FPT_TST_EXT.3	TSF Self-Test with Defined Methods
FPT_TUD_EXT.1	Trusted Update
FPT_TUD_EXT.2	Trusted Update based on certificates
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination (Refinement)
FTA_SSL.4	User-initiated Termination (Refinement)
FTA_TAB.1	Default TOE Access Banners (Refinement)
FTP_ITC.1	Inter-TSF trusted channel (Refinement)
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)
FTP_TRP.1/Admin	Trusted Path (Refinement)

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

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~~ **3**.

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~~ **3**.

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

FAU_GEN.1.1/VPN

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

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

FAU_GEN.1.2/VPN

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

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

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

Table 3: Auditable Events

SFR	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.

Table 3: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	None.	None.
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.

Table 3: Auditable Events

SFR	Auditable Events	Additional Audit Record Contents
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_TUD_EXT.2	Failure of update	Reason for failure (including identifier of invalid certificate)
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.
FAU_STG_EXT.2/LocSpace	None.	None.

Table 3: Auditable Events

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

6.1.1.3 FAU_GEN.2 User Identity Association

FAU_GEN.2.1

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

6.1.1.4 FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1

The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2

The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

6.1.1.5 FAU_STG_EXT.2/LocSpace Counting Lost Audit Data

FAU_STG_EXT.2.1/LocSpace

The TSF shall provide information about the number of overwritten audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

6.1.1.6 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2

The TSF shall be able to store generated audit data on the TOE itself. In addition

- The TOE shall consist of a single standalone component that stores audit data locally.

FAU_STG_EXT.1.3

The TSF shall overwrite previous audit records according to the following rule: **deletion of the oldest audit records, and generation of an audit record indicating that audit data overwriting has occurred**, when the local storage space for audit data is full.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (Refinement)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm:

- ECC schemes using 'NIST curves' P-256, P-384 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2.1[1][2]

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".

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

FCS_CKM.1.1/IKE [TD0723[3]]

The TSF shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a specified cryptographic key generation algorithm:

- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-384 and P-256

and

- no other key generation algorithms

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

[1] SFR text, Application Note and Evaluation Activities were modified by TD0580.

[2] SFR text was modified by TD0581.

[3] TD0723 was implemented.

6.1.2.4 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a single overwrite consisting of zeroes, a new value of the key;
- 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 single overwrite consisting of zeroes, a new value of the key;

that meets the following: No Standard.

6.1.2.5 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption)

FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in CBC, CGM and CTR 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 CTR as specified in ISO 10116.

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

FCS_COP.1.1/SigGen

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

- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes **256 and 384 bits**

that meet the following:

- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” P-256, P-384; ISO/IEC 14888-3, Section 6.4.

6.1.2.7 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-256, SHA-384, SHA-512 and message digest sizes 256, 384, 512 bits that meet the following: ISO/IEC 10118-3:2004.

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

FCS_COP.1.1/KeyedHash

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-384, implicit 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”.

6.1.2.9 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3

If a peer certificate is presented, the TSF shall not require client authentication, if the peer certificate is deemed invalid.

6.1.2.10 FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1.1

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

FCS_IPSEC_EXT.1.2

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

FCS_IPSEC_EXT.1.3

The TSF shall implement tunnel mode.

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-256 (specified in RFC 4106) and no other algorithm together with a Secure Hash Algorithm (SHA)-based HMAC no HMAC algorithm.

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.

FCS_IPSEC_EXT.1.6 [TD0657[4]]

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).

FCS_IPSEC_EXT.1.7

The TSF shall ensure that

- IKEv2 SA lifetimes can be configured by a Security Administrator based on
  - length of time, where the time values can be configured within **4 to 120** hours.

[4] The selections for this SFR were updated by TD0657.

FCS_IPSEC_EXT.1.8

The TSF shall ensure that

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on
  - length of time, where the time values can be configured within **2 to 48** hours.

FCS_IPSEC_EXT.1.9

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least **256 (for DH Group 19), and 384 (for DH Group 20)** bits.

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 DH group;
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash.

FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s)

- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and no other DH Groups according to RFC 5114.

FCS_IPSEC_EXT.1.12

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the 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.

FCS_IPSEC_EXT.1.13

The TSF shall ensure that all IKE protocols perform peer authentication using ECDSA that use X.509v3 certificates that conform to RFC 4945 and no other method.

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), no other reference identifier type.

6.1.2.11 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

6.1.2.12 FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

FCS_TLSC_EXT.1.1 The TSF shall implement TLS 1.2 (RFC 5246) and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and no other ciphersuites.

FCS_TLSC_EXT.1.2

The TSF shall verify that the presented identifier matches: the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.1.3

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also

- Not implement any administrator override mechanism.

FCS_TLSC_EXT.1.4

The TSF shall present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: secp384r1 and no other curves in the Client Hello.

6.1.2.13 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1

The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

6.1.2.14 FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

FCS_TLSS_EXT.1.1

The TSF shall implement TLS 1.2 (RFC 5246) and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and no other ciphersuites.

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1.

FCS_TLSS_EXT.1.3

The TSF shall perform key establishment for TLS using ECDHE curves secp384r1 and no other curves.

FCS_TLSS_EXT.1.4[5]

The TSF shall support no session resumption or session tickets.

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management (Refinement)

FIA_AFL.1.1

[5] The Application Note, TSS, Guidance and Test Evaluation Activities for this SFR were modified by TD0569.

The TSF shall detect when an Administrator configurable positive integer within **3 to 10** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2

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

6.1.3.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”;
b) Minimum password length shall be configurable to between **6** and **20** characters.

6.1.3.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- automated generation of cryptographic keys**, respond to ARP requests with ARP replies, packet forwarding through the IPsec tunnel, packet forwarding through BYPASS packet filtering table, ICMP echo reply (when configured in packet filtering table by the SA)**.

FIA_UIA_EXT.1.2

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

6.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism to perform local administrative user authentication.

6.1.3.5 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

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

6.1.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation[6]

FIA_X509_EXT.1.1/Rev

[6] Test 8 added to Test Assurance Activity by TD0527.

The TSF shall validate certificates in accordance with the following rules:

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

FIA_X509_EXT.1.2/Rev

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

6.1.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and HTTPS, TLS, and code signing for system software updates, no additional uses.

FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall accept the certificate, not accept the certificate.

ST Author Note: The mapping of the selections in FIA_X509_EXT.2.1 to FIA_X509_EXT.2.2 are as follows:

- *The "TLS" selection in element 1 maps to "accept the certificate" in element 2. "IPsec", while not a selection, also maps to "accept the certificate" in element 2.*
- *The "code signing for system software updates" selection in element 1 maps to "not accept the certificate" in element 2.*
- *The "HTTPS" selection in element 1 does not apply to element 2 - while the TOE uses it's own server certificate to authenticate itself to remote entities attempting to authenticate via the Trusted Path (HTTPS), the TOE does not (and is not required to) validate its own certificates.*

6.1.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

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

FIA_X509_EXT.3.2

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

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

FMT_MOF.1.1/ManualUpdate

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

6.1.4.2 FMT_MOF.1/Services Management of Security Functions Behavior

FMT_MOF.1.1/Services

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

6.1.4.3 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData

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

6.1.4.4 FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys

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

6.1.4.5 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to 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 <u>digital signature</u> capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
    - <u>Ability to manage the cryptographic keys;</u>
    - <u>Ability to configure the lifetime for IPsec SAs;</u>
    - <u>Ability to import X.509v3 certificates to the TOE’s trust store;</u>
    - <u>Ability to start and stop services</u>
    - <u>Ability to modify the behavior of the transmission of audit data to an external IT entity;</u>
    - <u>Ability to set the time which is used for time-stamps;</u>
    - <u>Ability to configure the reference identifier for the peer;</u>
    - <u>Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;</u>

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

FMT_SMF.1.1/VPN

The TSF shall be capable of performing the following management functions [

- Definition of packet filtering rules

- Association of packet filtering rules to network interfaces;
- Ordering of packet filtering rules by priority;
- No other capabilities.

6.1.4.7 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1

The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

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

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

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

FPT_SKP_EXT.1.1

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

6.1.5.3 FPT_STM.EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

The TSF shall allow the Security Administrator to set the time, obtain time from the underlying virtualization system.

6.1.5.4 FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests during initial start-up (on power on), to demonstrate the correct operation of the TSF: noise source health tests, **Cryptographic Known Answer Tests (KATs), TOE Firmware Integrity Check**.

6.1.5.5 FPT_TST_EXT.3 TSF Self-Test with Defined Methods

FPT_TST_EXT.3.1

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

FPT_TST_EXT.3.2

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

6.1.5.6 FPT_FLS.1/SelfTest Fail Secure (Self-Test Failures)

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].

6.1.5.7 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism and X.509 certificate prior to installing those updates.

6.1.5.8 FPT_TUD_EXT.2 Trusted Update Based on Certificates

FPT_TUD_EXT.2.1

The TSF shall check the validity of the code signing certificate before installing each update.

FPT_TUD_EXT.2.2

If revocation information is not available for a certificate in the trust chain that is not a trusted certificate designated as a trust anchor, the TSF shall not install the update.

FPT_TUD_EXT.2.3

If the certificate is deemed invalid because the certificate has expired, the TSF shall not accept the certificate.

FPT_TUD_EXT.2.4

If the certificate is deemed invalid for reasons other than expiration or revocation information being unavailable, the TSF shall not install the update.

6.1.6 Packet Filtering (FPF)

6.1.6.1 FPF_RUL_EXT.1 Rules for Packet Filtering

FPF_RUL_EXT.1.1

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

FPF_RUL_EXT.1.2 [TD0683[7]]

[7] TD0683 was implemented.

The TSF shall allow the definition of packet filtering rules using the following network protocols and protocol fields:

- IPv4 (RFC 791)
  - source address
  - destination address
  - protocol
- IPv6 (RFC 8200)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port
  - destination port

FPF_RUL_EXT.1.3

The TSF shall allow the following operations to be associated with packet filtering rules: permit and drop with the capability to log the operation.

FPF_RUL_EXT.1.4

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

FPF_RUL_EXT.1.5

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

FPF_RUL_EXT.1.6

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

6.1.7 TOE Access (FTA)

6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions,

- terminate the session

after a Security Administrator-specified time period of inactivity.

6.1.7.2 FTA_SSL.3 TSF-initiated Termination (Refinement)

FTA_SSL.3.1

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

6.1.7.3 FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4.1

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

6.1.7.4 FTA_TAB.1 Default TOE Access Banners (Refinement)

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.

6.1.8 Trusted path/channels (FTP)

6.1.8.1 FTP_ITC.1 Inter-TSF Trusted Channel (Refinement)

FTP_ITC.1.1

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

FTP_ITC.1.2

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

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for **Syslog connection from the Virtual Network Device to a Syslog server**.

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

FTP_ITC.1.1/VPN

The TSF shall be capable of using IPsec to provide a communication channel between itself and authorized IT entities supporting VPN communications that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2/VPN

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

FTP_ITC.1.3/VPN

The TSF shall initiate communication via the trusted channel for remote VPN gateways or peers.

6.1.8.3 FTP_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1.1/Admin

The TSF shall be capable of using HTTPS to provide a communication path between itself and authorized remote Administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

FTP_TRP.1.2/Admin

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

FTP_TRP.1.3/Admin

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

6.2 Security Assurance Requirements

This Security Target is conformant with the assurance requirements specified in the cPP.

Table 4: Assurance Requirements

Assurance Class	Assurance Component
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

7. TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following Security Functions:

- Security Audit
- Communication
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

7.1 Security Audit

7.1.1 Audit Data Generation

The TOE generates and locally stores audit records for the following auditable events:

- Start up and shutdown of the audit function (as startup and shutdown messages for the OS, since logging may not be started or stopped independently of the TOE startup and shutdown).
- All security administrative actions, including:
  - Administrative login and logout, including username
  - Security related configuration changes (in addition to the information that a change occurred, what has been changed is also logged)
  - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference is logged as described below)
  - Resetting passwords (name of related user account is logged)
  - Starting and stopping services
  - All commands entered during administrative sessions
  - All events specified in Table 3 ‘Auditable Events’.

Auditing is provided by the TOE Operating System subsystems which is configured to store the audit logs locally and transmit them to the administrator-configured remote Syslog service over ‘syslog RFC 5424’ protocol tunneled through the TLSv1.2 protocol. Local audit logs are stored in the TOE’s underlying file system. Only an authorized and authenticated administrator can view or delete log files stored on the TOE, via the CLI only. Such actions require being first authenticated as an authorized security administrator via the local console.

The TOE only defines one user role, and that is of the “Security Administrator” role.

For each audit event, the TSF associates the audit event with the identity of the user that caused the event. Each audit log contains a date-and-time stamp of the event, the type of event, subject identity, and outcome (success or failure) of the event. Additional information, if available, is also logged according to the details provided in [AGD].

For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the following is logged to identify the relevant key:

- Event: Key pair/CSR generation:
    - Text that states that a cryptographic key was generated for service X, where X is one of the following services:
        - IPsec
            - Identified in audit records as ‘ipsec’
        - Syslog
            - Identified in audit records as ‘syslog’
        - RMI
            - Identified in audit records as ‘rest_api’
- Event: Import of cert:
    - Service ID (identified as ‘ipsec’, ‘rest_api’, or ‘syslog’)
- Event: Key Deletion:
    - File type (identified as ‘key’, ‘cert’, ‘csr’), and service (identified as ‘ipsec’, ‘rest_api’, or ‘syslog’)
- Event: Trust Store Change:
    - Import:
        - Service, CA type and overwrite vs. new import
    - Delete:
        - Service, cert/CA type

Only one private key/CSR, one Root CA cert and one intermediate CA cert per each service are allowed in the vND

Regarding packet filtering and auditing; the TOE has several components that are involved in network traffic processing as depicted in Figure 2 below. First, the ingress network protocol stack handles network traffic and if unable to service all arriving packets, will discard packets that cannot be processed. The next component that handles IPsec traffic is the Firewall (FW) component. The FW functions are positioned at the Red and the Black network interfaces of the TOE. When presented with excessive network traffic, the FW component would drop the packets it cannot process. The next component in the IPsec traffic processing chain is the IPsec VPN function that implements part of the traffic processing policy. When presented with excessive network packets it cannot handle, it may drop the packets or fail secure. This fail secure functionality is provided by a watchdog service that restarts the IPsec process if it has failed. The FW function on the egress network interface would not permit egress of packets that have “Protect” IPsec traffic processing policy from leaving the TOE unless encapsulated inside an IPsec ESP packet.

While the packets might be dropped and not logged by the TOE when the TOE is presented with excessive amount of traffic, no packets will leave the TOE unless they can be adjudicated by the packet processing policy in effect.

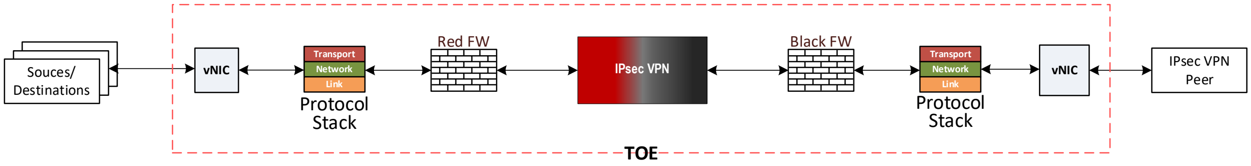

Figure 2 – TSF Components for Network Traffic Processing

FAU_GEN.1; FAU_GEN.2

7.1.2 Audit Storage

The TOE stores audit records locally as well as transmits them to an external, administrator-configurable remote audit server over TLS. The locally stored audit records are unable to be viewed except by an authenticated, authorized administrator, via the CLI only.

FAU_STG.1

The TOE is a standalone TOE (virtual network device). The transmission of data to an external server is real-time. The TOE allocates a total of 10 MB for local storage of audit records. This is achieved by creating ten (10) files each with a total size of one (1) megabyte. The TSF writes audit records to one of these files then when the file reaches one megabyte in size, the TSF begins to write to the next file. Once the tenth file is full, the TSF will then delete the oldest file and begin writing all subsequent audit records to the new file. The TSF repeats this audit log rotation function as needed.

All stored audit logs are protected against unauthorized deletion, modification, or viewing. To view these files, a user must be authenticated as an authorized administrator via the CLI.

FAU_STG_EXT.1

When the oldest audit record file (the tenth file) is deleted, the TSF creates an audit record which states the number of audit records that were present in the oldest audit record file that was deleted by the audit log rotation TSF. This audit record is placed in the newest audit record file created by the audit log rotation TSF. This functionality is enabled by default and is not configurable nor can it be turned off.

FAU_STG_EXT.2/LocSpace

7.2 Cryptographic Support

7.2.1 Cryptographic Key Generation

Table 5 below identifies the CAVP validated cryptographic algorithms and their associated CAVP certificate identifier, utilized by the TSF in the evaluated configuration:

Table 5 - CAVP Certificates

Function	SFR	Purpose/Association	Cert No.
Cryptographic Key Generation (asymmetric)	FCS_CKM.1.1; FCS_CKM.1.1/IKE	**Key Gen for Authentication:** • IPsec Authentication via x509 Certificates: ○ ECDSA: ▪ ECC FIPS PUB 186-4 P-256, and P-384 curves • TLS Authentication via x509 Certificates: ○ ECDSA: ▪ ECC FIPS PUB 186-4 P-384 curve **Key Gen for Establishment:**	**A4427**

		• **IPsec** o Diffie-Hellman groups 19, 20: ▪ ECC FIPS PUB 186-4 for P-256, and P-384 curves • **TLS:** o ECDHE Key exchange: ▪ ECC FIPS PUB 186-4 P-384 curve	
Cryptographic Key Establishment	FCS_CKM.2.1	**IPsec:** • Diffie-Hellman groups 19, and 20 (ECC CDH): o NIST SP 800-56A Revision 3, ECC CDH Scheme [(Cofactor) Ephemeral Unified Model] for P-256, and P-384 curves **TLS:** • ECDHE Key exchange: o NIST SP 800-56A Revision 3, ECC CDH Scheme [(Cofactor) Ephemeral Unified Model] for P-384 curve	**A4427**
AES Data Encryption/ Decryption	FCS_COP.1.1/DataEncryption	**DRBG:** • AES-CTR-256 **IPsec IKEv2:** • AES-CBC-256 • AES-GCM-256 **IPsec ESP:** • AES-GCM-256 **TLS**: • AES-GCM-256	**A4427**
Signature Generation and Verification	FCS_COP.1.1/SigGen	**Signature generation and verification for TLS Authentication via x509 Certificates:** • ECDSA: o ECC FIPS PUB 186-4 using P-384 curve	**A4427**

		Signature generation and verification for IPsec Authentication via x509 Certificates: • ECDSA: ○ ECC FIPS PUB 186-4 using P-256 and P-384 curves **Signature verification for TOE firmware validation (FPT_TST_EXT.1.1) and Trusted Update (FPT_TUD_EXT.1.3) security functionality:** • ECDSA: ○ ECC FIPS PUB 186-4 using P-384 curve	
Cryptographic Operation (Hash Algorithm)	FCS_COP.1.1/Hash	**SHA-256; SHA-384**: • IPsec: ○ IKEv2: ▪ Pseudorandom Function (PRF) used to generate cryptographic keys for both IKE SA and CHILD SA cryptographic algorithms [HMAC-SHA-384 (SHA-384) when AES-GCM-256 or AES-CBC-256 is negotiated] ▪ ECDSA P-256 (SHA-256), and P-384 (SHA-384) Authentication (sig.gen, sig.ver) ○ ESP: ▪ No hash since AES-GCM-256 provides authenticated encryption • TLS: ○ TLS KDF: ▪ SHA-384 (HMAC-SHA-384) ○ Keyed Hashing:	**(SHA-256; SHA-384) A4427** **(SHA-512) A4428**

		▪ HMAC-SHA-384 (SHA-384) o Authentication (sig.gen, sig.ver): ▪ ECDSA P-384 (SHA-384) **SHA-512:** • TOE Firmware Verification and Trusted Update: o SHA-512 • Passwords: o SHA-512	
Cryptographic Operation (Keyed Hash Algorithm)	FCS_COP.1.1/KeyedHash	**HMAC-SHA-384**: • IPsec: o IKEv2: ▪ HMAC-SHA-384: Pseudorandom Function (PRF) used to generate cryptographic keys for both IKE SA and CHILD SA cryptographic algorithms • TLS KDF: o HMAC-SHA-384 (SHA-384) HMAC-SHA-384: • 1024-bit block size • Digest/Output Length of 512-bits • Key Length of 512-bits	**A4427**
Random Bit Generation	FCS_RBG_EXT.1	CTR_DRBG (AES) for both IPsec and TLS trusted path/channels.	**A4427**

The TSF supports P-256 and P-384 curve for signature generation and verification of ECDSA-based x509 certificates for authentication purposes for IPsec protocol connections while TLS only supports the P-384 curve.

The TOE complies with NIST FIPS 186-4 Appendix B.4 (ECC Key Pair Generation) and B.4.2, (Key Pair Generation by Testing Candidates), for the generation of ECC cryptographic key pairs. The TOE implements all “shall” and “should” statements and does not implement any ‘"shall not" " or "should not" statements from these two sections. Regarding the one “should” statement in B.4.2; if an error is encountered during the generation process, the invalid values are returned – specifically, the TSF returns an ERROR indication with the values of Invalid_d and Invalid_Q.

FCS_CKM.1; FCS_CKM.1.1/IKE

The supported key establishment schemes supported by the TSF are described in the table below:

Table 6 – Key Agreement Schemes and Associated Information

Scheme	SFR	Service	Key Sizes
ECDHE	FCS_TLSS_EXT.1 & FCS_TLSC_EXT.1	Remote Administration (TLSS) and Remote Audit Record Transport (TLSC)	P-384 curve
Diffie-Hellman (Groups 19 and 20)	FCS_IPSEC_EXT.1.9	IPsec (IKEv2) key exchange	P-256 & P-384 curves

FCS_CKM.2

The cryptographic keys and key material associated with the TSF are described in the table below (IKE CSP identifiers are indicated using the notation provided in RFC 5996; TLS CSP identifiers are indicated using the notation provided in RFC 5246):

Table 7 – Cryptographic Keys and Key Material Associated with the TSF

Item	Description	Storage Location	Timing and Method of Zeroization
TLS CSPs (plaintext) in depth			
TLS pre_master_secret	Generated during key agreement for ECDHE cipher suite	RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
client_write_MAC_key *(not generated by TSF due to GCM ciphersuite)*		RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject

			• Overwritten with 0x00
server_write_MAC_key *(not generated by TSF due to GCM ciphersuite)*		RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
client_write_key		RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
server_write_key		RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
client_write_IV		RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00

server_write_IV		RAM	- Automatically and immediately upon termination of the established SA that utilized the key in subject - Overwritten with 0x00
X.509 certificate key pairs for the following TSF: - IPsec key pair for authentication to remote IPsec peer - TLS Client key pair for mutual authentication to remote audit server - TLS Server key pair for RMI service identity	ECDSA private key is generated when the CSR is generated by the Security Administrator. The private key is persistently stored. The private key is loaded into RAM during session establishment, when signature generation within the session establishment process occurs.	RAM & Persistent storage	RAM: - Automatically and immediately upon termination of the established Trusted Channel/Path session that utilized the key in subject; overwritten with 0x00 Persistent storage: - When the keypair is deleted by the Security Administrator, the data is overwritten with 0x00 - When the keypair is replaced by the Security Administrator, the data is replaced with the new value of the key pair - When the SA issues a command to restore factory default settings

			of the RMI, the entire RMI Trust Store is overwritten with 0x00 - When the SA issues a command to destroy all cryptographic keys on the system; overwritten with 0x00 *(NOTE: the x.509 certificates associated with the Trusted Update TSF are never deleted, even if this command is run).*
IPsec CSPs (plaintext) in depth			
Diffie-Hellman shared secret (g^ir)	Generated during IKE SA establishment	RAM	- Automatically and immediately upon termination of the established Trusted Channel/Path session that utilized the key in subject; overwritten with 0x00
SKEYSEED	used to calculate the seven other CSPs described below	RAM	- Automatically and immediately upon termination of the established SA that utilized the key in subject - Overwritten with 0x00
SK_d	Used when deriving new keys for the IPsec	RAM	- Automatically and

	Child SAs created by a particular IKE_SA		immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
SK_ai & SK_ar	IKE_SA HMAC keys (initiator and receiver keys) **NOTE: while these keys may be present in RAM since they are calculated by the IKE SA establishment algorithms regardless of the encryption algorithm that is ultimately negotiated - these keys are not used when AES_GCM is the encryption algorithm negotiated during IKE SA establishment. These keys are used for HMAC data integrity and authentication when AES_CBC is the encryption algorithm negotiated during IKE SA establishment.*	RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00
SK_ei and SK_er	IKE_SA AES encryption/decryption (initiator and receiver keys) *NOTE: When AES_GCM is the negotiated encryption algorithm, there is a salt for each of these keys. For AES_GCM_256, the salt is 32 bits in length. The salt is considered a CSP and is handled by the TSF identically as the associated AES key.	RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00

SK_pi and SK_pr	IKE_SA Authentication keys (initiator and receiver keys)	RAM	• Automatically and immediately upon termination of the established SA that utilized the key in subject • Overwritten with 0x00

Upon a command from an SA, the TSF can perform key zeroization. This command deletes all device private PKI keys and device certificates persistently stored on the device. The KMAT data is overwritten with all zeros and the file metadata is removed. In cases of power loss during this KMAT delete operation, it is possible for KMAT or remnants of KMAT to remain present in virtual disk storage. If a power loss or forced VM shutdown occurs during performance of this KMAT deletion operation, repeating the delete operation command would remove any remaining KMAT data from persistent storage.

When new Key Material (KMAT) is loaded into the TOE, the content of the prior KMAT is overwritten with the new KMAT. This TSF applies only to the KMAT that is configurable by the SA – this includes persistent authentication-keys that are generated by, and/or installed by, the security administrator for the IPsec, TLS client and TLS server TSF.

The TOE does not destroy KMAT under other circumstances, with the exception of the Trusted Update TSF. When the Trusted Update is performed and the Security Administrator selects to remove all existing configuration, the KMAT and all other configuration is destroyed, by single overwrite of zeros, along with the prior software version's File System (FS). The new software version when installed recreates the new FS.

The above holds true for the operational environment described in this ST, which utilizes a spinning disk hard drive; however, it is important to note that if a Solid State Drive (SSD) that implements a wear leveling technology is used, it is possible that remnants of device's KMAT are left on the SSD after KMAT deletion – therefore, it is imperative that spinning disk hard drives are used to ensure enforcement of the SFRs.

All of the key destruction methods described above utilize Linux OS system API to perform secure cryptographic key destruction.

FCS_CKM.4

7.2.2 Cryptographic Operations

The key sizes and mode of operation for the TSF's utilization of the AES algorithm for data encryption/decryption is described in the table below:

Table 8 – Key Size and Mode of Operation for AES

Algorithm	Associated SFR	Uses and Details
AES Data Encryption/ Decryption	FCS_COP.1.1/DataEncryption	**IPsec IKEv2:** • AES-CBC-256

		○ Mode = CBC; Key Size = 256-bits • AES-GCM-256 ○ Mode = GCM; Key Size = 256-bits **IPsec ESP:** • AES-GCM-256 ○ Mode = GCM; Key Size = 256-bits **TLS**: • AES-GCM-256 ○ Mode = GCM; Key Size = 256-bits

FCS_COP.1/DataEncryption

The algorithms and key sizes for the TSF's utilization of signature generations and verification is described in the table below:

Table 9 – Algorithms and Key Sizes for Digital Signature Services

Service	Associated SFR	Uses and Details
Signature Generation and Verification	FCS_COP.1.1/SigGen	**Signature generation and verification for IPsec Authentication via x509 Certificates:** • ECDSA: ○ ECC FIPS PUB 186-4 using P-256, and P-384 curves **Signature generation and verification for TLS Authentication via x509 Certificates:** • ECDSA: ○ ECC FIPS PUB 186-4 using the P-384 curve **Signature verification for TOE firmware validation (FPT_TST_EXT.1.1) and Trusted Update (FPT_TUD_EXT.1.3) security functionality:** • ECDSA: ○ ECC FIPS PUB 186-4 using P-384 curve

FCS_COP.1/SigGen

The algorithms and key sizes for the TSF’s utilization of cryptographic hashing services is described in the table below:

Table 10 – Algorithms and Key Sizes for Cryptographic Hashing Services

Service	Associated SFR	Uses and Details
Cryptographic Operation (Hash Algorithm)	FCS_COP.1.1/Hash	**SHA-256, SHA-384**: • IPsec: o IKEv2: ▪ Pseudorandom Function (PRF) used to generate cryptographic keys for both IKE SA and CHILD SA cryptographic algorithms [HMAC-SHA-384 (SHA-384)] ▪ ECDSA P-256 (SHA-256), and P-384 (SHA-384) Authentication (sig.gen, sig.ver) ▪ Data authentication and integrity uses HMAC-SHA-384 (SHA-384)]; note that the SHA-384 primitive is still used even when truncation of the output of the HMAC for authentication and data integrity services occurs] o ESP: ▪ No hash since AES-GCM-256 provides authenticated encryption • TLS: o TLS KDF: ▪ SHA-384 (HMAC-SHA-384) o Keyed Hashing: ▪ HMAC-SHA-384 (SHA-384) o Authentication (sig.gen, sig.ver): ▪ ECDSA P-256 (SHA-256) and P-384 (SHA-384)

		• TOE Firmware Verification and Trusted Update: ○ SHA-512 • Passwords: ○ SHA-512

FCS_COP.1/Hash

The algorithms and key sizes for the TSF's utilization of Keyed Hashing is described in the table below:

Table 11 – Algorithms and Key Sizes for Cryptographic Keyed Hashing Services

Algorithm	**Associated SFR**	**Uses and Details**
Cryptographic Operation (Keyed Hash Algorithm)	FCS_COP.1.1/KeyedHash	**HMAC-SHA-384**: • IPsec: ○ HMAC-SHA-384: Pseudorandom Function (PRF) negotiated in IKEv2, used to generate cryptographic keys for the cryptographic algorithms used in both the IKE SA and CHILD SA. ○ HMAC-SHA-384: for data authenticity/integrity services used in both the IKE SA and CHILD SA. • TLS KDF: ○ HMAC-SHA-384 (SHA-384) HMAC-SHA-384: • 1024-bit block size • Digest/Output Length of 384-bits • Key Length of 384-bits

FCS_COP.1/KeyedHash

7.2.3 HTTPS Protocol

RFC 2818 (HTTP Over TLS) describes how to use TLS to secure HTTP connections over the Internet. Details of TSF behavior with regards to the following specifications listed in RFC 2818 are as follows:

- Section 2.1, Connection Initiation:
  - TSF meets described behavior; no deviation
- Section 2.2, Connection Closure:
  - TSF sends TLS closure alert when terminating an HTTPS connection.
  - Session reuse is not implemented by the TSF
  - TSF meets described behavior; no deviation
- Section 2.2.1, Client Behavior:

    - The TSF acts as the HTTPS Server, therefore this section does not pertain to the TSF
- Section 2.2.2, Server Behavior:
    - The TSF does not support TLS session resumption
    - The TSF will attempt to initiate an exchange of closure alerts with the client before closing the connection
- Section 2.3, Port Number:
    - TSF utilizes TCP port 443 to listen for incoming HTTPS connections
- Section 2.4, URI Format:
    - TSF supports and requires the “https://” URI protocol identifier prefix for incoming HTTPS requests
- Section 3.1, Server Identity:
    - Pertains to client behavior, not applicable to the TOE
- Section 3.2, Client Identity:
    - The TSF does not support mutual authentication with regards to the HTTPS TSF

FCS_HTTPS_EXT.1

7.2.4 IPsec Protocol

The TSF implements IPsec as specified in RFCs 3602, 4106, 4301, 4303, 4868, 5114, 5282, 5996. The TSF only supports IKEv2 and ESP connections operating in tunnel mode.

The TSF uses the Linux iptables service to perform IPsec Security Policy Database (SPD) as described in Section 7.4.2. Packets not matching a rule are dropped by a hard-coded final rule.

The TSF allows three actions to be assigned to packet rules:

- Allow
    - protect the packet with IPsec
- Drop
    - drop the packet with no further processing
- Bypass
    - allow the packet to flow through the TOE with no protection
- Log
    - Generate audit record for the packet matching a rule

The TSF enforces SPD entries in the order the administrator configures the packet processing rules.

The TSF supports only the use of the AES-GCM-256 algorithm for encryption and message authentication for the IPsec ESP protocol.

The TSF supports AES-GCM-256 or AES-CBC-256 to encrypt the IKEv2 payload and HMAC-SHA-384 (only when CBC-based cipher is negotiated) to authenticate the IKEv2 payload.

The TSF supports truncation of the output of the HMAC algorithm for authentication and data integrity services using the HMAC_SHA_384_192 algorithm for the IKE SA, only when paired with the AES-CBC cipher. When truncation occurs, the output of the HMAC is 192-bits in length. The output is never truncated when HMAC is used as the PRF for generating session keys. When AES-GCM cipher is negotiated, an HMAC for authenticity and data integrity is not used since AES-GCM provides the necessary data origin authentication and integrity verification services.

The TSF supports IKE_SA lifetime configuration from 4 hours to 120 hours whereas the CHILD_SA lifetime configuration is configurable from 2 hours to 48 hours. By default, if no packets are processed by the CHILD_SA within the configured SA lifetime, the SA is closed. In this mode, the IKEv2 SA lifetime like acts as an idle timeout value. If any packets were processed through the IKEv2 SA tunnel, the tunnel is re-keyed instead.

The TSF supports DH groups 19 and 20 for use in IKEv2. These groups are not configurable by the administrator. The TSF will negotiate the algorithms in the following order: 19 then 20. The TSF generates the following ephemeral private key (x) sizes used in Diffie-Hellman based on the negotiated group.

Table 12: Diffie-Hellman Key Sizes

Group	Private Key Size
19	256-bits
20	384-bits

The TSF negotiates the allowed groups with the peer in the IKEv2 exchange.

The TSF generates and proposes nonces that are 256 bits long. The nonces are used in the IKEv2 key exchange for all cipher suites and are generated by the DRBG as defined in FCS_RBG_EXT.1. A 256-bit nonce is sufficient to meet the security strengths of all configurable and supported Diffie-Hellman groups, as well as being at least 128 bits and half the strength of the negotiated PRF hash. Because nonces may be created prior to the DH group being chosen, this posture allows the TSF to maximize cryptographic security across all possible key agreement parameters.

The symmetric key size for AES in the IKE_SA and CHILD_SA will always be the same since the TSF supports only 256-bit key size for AES. This ensures that the symmetric key size negotiated for a CHILD_SA is 'less than or equal' to the key size negotiated for the IKEv2 SA. If a peer attempts to negotiate a CHILD_SA with a key size that has not been configured on the TSF, the connection attempt will fail with a cipher-suite mismatch.

The TSF supports NAT traversal automatically, and is automatically applied if NAT is detected. This is not configurable by the administrator.

The TSF authenticates peers as describe in Section 7.3.7**Error! Reference source not found.** using X.509 certificates with any of the following algorithm/keysize combinations:

- ECDSA P-256
- ECDSA P-384

The TSF only establishes a trusted channel to the remote VPN peer if the presented identifier in the received peer certificate matches the security-administrator configured reference identifier, where the presented and reference identifiers are a Distinguished Name (DN). The TSF supports the following Relative Distinguished Names (RDN):

- Common Name (CN)
- Organization (O)
- Organizational Unit (OU)
- Locality (L)
- State (S)
- Country (C)

7.2.5 Random Bit Generation

Random bit generation is provided by the CTR_DRBG(AES) [256-bit] algorithm, in accordance with ISO/IEC 18031:2011A. This functionality is not configurable, and there are no other cryptographic engines provided in the TOE. The TSF utilizes Intel's on-chip entropy source (RDSEED) as the seed data into the DRBG, which is then utilized by the TSF for generating random numbers which are utilized by the cryptographic operations of the TSF when generating cryptographic CSPs. The DRBG is seeded with 640 bits from the entropy source. This provides the DRBG with at least 256-bit of entropy (min-entropy), will providing a 128 bit nonce in addition to at least 256 bits of entropy.

FCS_RBG_EXT.1

7.2.6 TLS Client Protocol Without Mutual Authentication

The TSF implements a TLSv1.2 client according to RFCs 4492, 5246, 5289, and 6125. The TSF supports the following ciphersuite:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Ciphersuites are not user-configurable.

The TSF sends the Supported Elliptic Curves extension with secp384r1 in its TLS Client Hello message. The TSF does not allow for the configuration of elliptic curves supported for TLS Client functionality.

The TSF verifies the remote TLS server's certificate is valid as described in Section 7.3.6.

The SA must provide both a DNS entry and an IP address configuration entry for the target remote syslog server. The domain name is used for establishing the reference identifiers to authenticate using X.509 certificates.

When the remote Syslog server destination is configured, the TSF automatically generates DNS-ID reference identifiers containing the configured DNS name. When the certificate presented by the syslog server contains the SAN extension the TSF compares the DNS-ID to the DNS Name SAN fields. Otherwise, the TSF compares the reference identifier to the CN field(s). In both cases, the TSF performs the comparison as specified in Section 6.4 of RFC 6125. The TSF does not support wildcards in the domain labels listed in the peer certificate(s).

If the certificate validation or reference identifier matching fails, the TSF does not establish the connection. If the certificate revocation status cannot be determined (e.g. CDP unreachable), the TSF will accept the connection assuming all other validation checks pass.

The TSF will reject a connection attempt if the peer certificate contains only a CN and/or SAN with IP address identifier. If the certificate validation or reference identifier matching fails, the TSF does not establish the connection.

The TSF does not support certificate pinning.

The TSF sends its X.509 certificate in a Client Certificate message and signs a Certificate Verify message using the private key associated with the X.509 certificate to authenticate itself to the server.

The TSF enforces canonical format as described in RFC 3986 for IPv4.

FCS_TLSC_EXT.1

7.2.7 TLS Client Support for Mutual Authentication

The TSF will send its TLS client certificate when a remote audit server sends a client certificate request TLS message. The TSF sends the client certificate that is associated with the certificate requested by the remote audit server.

FCS_TLSC_EXT.2

7.2.8 TLS Server Protocol Without Mutual Authentication

The TSF implements a TLSv1.2 server according to RFCs 4492, 5246, 5289, and 6125. The TSF supports the following ciphersuite:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Ciphersuites are not user-configurable.

The TLS server provides a trusted path (RMI) for performing administrative functions, during the course of which TLSv1.2 mechanisms are exercised.

The TOE denies all connection requests from TLS version 1.1 or older, and SSLv3.0 and older. Only TLSv1.2 connections are supported. When a client requests an unsupported version of TLS, the TOE rejects the connection attempt by sending "handshake failure" and "invalid protocol version" error responses to the client. This behavior is enforced in the OpenSSL configuration files, which are configured at the factory.

For ECDHE key agreement, the TOE will negotiate key establishment using NIST curve secp384r1. This is enforced by the TSF and is not configurable.

FCS_TLSS_EXT.1

7.3 Identification and Authentication

7.3.1 Authentication Failure Management

For each remote administrator, each successive unsuccessful authentication attempt is tracked using a counter. This counter is reset upon a single successful authentication attempt for the specific account. Each authentication failure and success event is logged.

The remote administrator is prevented from successfully logging into the TOE by the system after a configured number of unsuccessful authentication attempts is reached. Once the number of unsuccessful attempts to authenticate reaches the configured value (from 3 to 10 attempts), the user will be locked out for a security administrator configurable time interval (from 60 to 3600 seconds). The ability to login through the RMI will be restored by the system after the configured lockout time has elapsed. The security administrator account is configured to not be subjected to account locking when logging in locally via the local console. This is done to ensure that authentication failures by a remote administrator cannot lead to a situation where no administration access is available, either permanently or temporarily.

FIA_AFL.1

7.3.2 Password Management

Passwords created for TOE authentication may be composed of all printable ASCII characters in UTF-8 formatting. For local console and RMI sessions, the 'password minimum length' for administrative passwords is configurable by changing the TSF's password policy and may be configured to be between 6 and 20 characters. Such password policies are managed by the authenticated administrator and are enforced by the login module.

Passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".

FIA_PMG_EXT.1

7.3.3 User Identification and Authentication

The supported authentication mechanisms are via the trusted path provided by TLS/HTTPS protected interface (RMI), or the local console.

For TLS/HTTPS remote administrative sessions, an administrator must connect to the TOE using a TLS client (web browser or other TLS client capable of establishing an HTTPS session), which will negotiate a secure connection using the supported cryptographic cipher. During TLS remote administrative sessions, the administrator must authenticate using username and password credentials. A successful login will be denoted by obtaining an authentication token for the RMI TSF. Unsuccessful logons are identified with a message stating that the username and/or password were incorrect.

Administrators authenticate by providing their username and password. Passwords are stored salted and hashed (SHA-512) on the TOE.

For Local administrative sessions, administrators authenticate by providing their username and password at the logon prompt. Successful authentication is denoted by obtaining a command prompt, while unsuccessful authentication results in a prompt to reauthenticate. No administrative actions or functions are available prior to successful identification/authentication.

Only those pre-authentication functions described below are permitted before forcing the user or non-TOE entity to authenticate:

- Display the warning banner in accordance with FTA_TAB.1;
- respond to ARP requests with ARP replies
- automated generation of cryptographic keys for the TLS Server, and TLS Client TSF
- packet forwarding through the IPsec tunnel
- packet forwarding through BYPASS packet filtering table
- ICMP echo reply (when configured in packet filtering table by the SA)

FIA_UIA_EXT.1

7.3.4 Password-based Authentication Mechanism

The TOE provides a local password-based authentication mechanism to identify and authenticate users before allowing them to perform actions or execute commands on the TOE for both local and remote administrative sessions. The TSF stores authentication data for remote and local console sessions in non-plaintext form (salted and hashed, SHA-512) in the underlying filesystem.

FIA_UAU_EXT.2

7.3.5 Protected Authentication Feedback

During local console authentication, the TOE obscures password input by failing to echo back the characters entered by the user. This protects the administrator authentication data by revealing neither the content nor any related data regarding the administrator credentials (such as length or complexity).

FIA_UAU.7

7.3.6 X.509 Certificate Validation

Certificate validation occurs during the following events listed below:

- when a CA is imported/installed on the TOE
- when a signed CSR is imported/installed on the TOE
- when the TSF receives an X.509 certificate from a remote TLS Server peer
- when the TSF receives an X.509 certificate from a remote IPsec peer
- when the TOE Trusted Update functionality is executed

For certificate revocation status checking, CRLs are downloaded over HTTP only and using IP or FQDN identifier (though the TSF does not perform DNS resolution and thus local name-to-IP mapping configuration is required). Certificate revocation status checking takes place as a part of every certificate validation attempt. This check applies to every certificate in the chain of trust, except for the root certificate, which is explicitly trusted by way of installation in the TOE trust store.

For TLS Client and IPsec connections, if a CRL fails to download, the connection is still accepted given all other validation checks pass. Certificate revocations in a valid and successfully downloaded CRL are appropriately processed by the TSF in that the TSF will reject connection attempts when a revoked certificate is identified.

For the Trusted Update TSF, if a CRL fails to download (e.g. CDP is not reachable), the update attempt is rejected. Certificate revocations in a valid and successfully downloaded CRL are appropriately processed by the TSF in that the TSF will reject update attempts when a revoked certificate is identified.

For the TOE's own certificates (TLS Server certificate for Trusted Path; IPsec device certificate for Trusted Channel, TLS Client Certificate for Trusted Channel) the TSF validates these certificates upon their import into the TSF trust stores. If a CRL fails to download during this validation step, the certificates are still accepted/imported given all other validation checks pass. Certificate revocations in a valid and successfully downloaded CRL are appropriately processed by the TSF in that the TSF will reject connection attempts when a revoked certificate is identified.

If a CRL is signed by a Certificate Authority which does not contain the CRL Sign bit, the TSF rejects the CRL information and rejects the connection attempt.

The TOE validates x509v3 certificates according to the validation rules described in RFC 5280. The validation rules applicable to the TOE are as follows:

1. Current date between the "Valid from" and "Valid to" dates
2. Revocation Status of the certificate as specified in the CDP field of the certificate:
    - RFC 5280 Section 6.3 supporting RFC 5759 Section 5 (recursion reference to RFC 579 Section 4)
3. If the certificate is used for specific purposes, additional checks are performed on extendedKeyUsage:
    - Remote audit 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.
    - 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.
    - 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.
4. The certificate path is valid:
    - The certificate is signed by a certificate that:
        - Passes all of the certificate validation rules
        - Has the 'certificate signing' key-usage extension
        - Has basic constraints CA=True or the certificate is signed by a trusted Root CA

    ;or

    - The certificate chain of trust (itself, any intermediary CA certificates) chain to a trusted root CA.

For TLS and IPsec TSF, the TOE only requires that a root CA or Trust Anchor is installed into the TOE's trust store, which allows the rest of the chain of trust to be completed by receipt from the remote peer. If a complete chain of trust is not installed in the TOE and the peer does not provide the details to complete the chain of trust, the TOE will reject connection attempts in these cases.

FIA_X509_EXT.1/Rev

7.3.7 X.509 Certificate Authentication

The TSF has a separate Trust Store for each of the following TSF:

- RMI
- IPsec
- TLS client to remote Syslog server

Note that for the Trusted Update TSF, while having it's own trust store, it is not configurable.

For IPsec, the security administrator configures the certificate the TOE will use to identify itself to its IPsec peer. The TSF uses the certificate presented in the IKE authentication phase to identify and authenticate the IPsec peer.

For the RMI TSF, the security administrator configures the certificate the TOE will use to identify itself to a peer TLS client.

For TLS/Syslog, the administrator configures the certificate the TOE will use to identify itself to the TLS server. The TSF uses the certificate presented by the TLS server to identify and authenticate the remote audit server.

For certification revocation status to be obtained by the TSF, the appropriate CRL Distribution Point(s) must be accessible from the TOE's management network interface.

For both the Trusted Channel to a remote server, and the IPsec TSF, the TOE does not need to have a full certificate chain installed internally to successfully validate certificates. When a certificate chain is completed by the peer providing an intermediate CA certificate in the respective protocol exchanges, and this intermediate CA can be traced up to the device-stored Trust Anchor for the service, assuming no other certificate check failures, the chain will validate successfully.

FIA_X509_EXT.2

7.3.8 X.509 Certificate Requests

The TSF allows the administrator to generate CSRs that contain:

- Public Key
- Common Name
- Country
- Email
- Organization
- Organization Unit

The administrator may configure the common name and Subject Alternative Name fields, and these fields do not contain any information that is outside the control of the administrator.

FIA_X509_EXT.3

7.4 Security Management

7.4.1 Management of Security Functions Behavior

The TSF does not allow administrative actions to be performed prior to successful identification and authentication of the security administrative user. Once the security administrative user is successfully authenticated, the TSF grants the user access to a restricted command-line shell. This shell restricts the security administrative user to only those commands required for administering the TSF while preventing users from running general-purpose Linux commands.

The TSF enforces these restrictions by restricting administrators to a restricted menu-driven management interface. Any changes to the system configuration or stored data can be performed only by an identified and authenticated security administrator.

The TSF restricts the following functions to identified and authenticated administrators (the management interface is also identified below):

- Manage X.509 certificates (import/remove certificates; designate trust anchors)
  - RMI
- Configure IKEv2 SA lifetimes
  - CLI
- Configure IPsec peer identities
  - RMI
- Ability to configure the access banner
  - CLI
- Configure remote audit server peer identities
  - RMI
- Generate CSR (ECDSA key pair)
  - RMI
- Manage security administrator password
  - CLI
- Configure minimum password length
  - CLI
- Configure the remote administrator inactivity timeout
  - CLI
- Configure the local administrator inactivity timeout
  - CLI
- Manage the failed authentication lockout threshold
  - RMI
- Configure Syslog server connectivity
  - RMI
- Initiate an update to the software
  - RMI
- Set the system date and time
  - Both CLI and RMI
- Definition of packet filtering rules
  - IPv4
    - Both CLI and RMI
  - IPv6 (note that only DROP and BYPASS rules are supported for IPv6)
    - CLI
- Association of packet filtering rules to network interfaces
  - CLI

- Ordering of packet filtering rules by priority
    - Both CLI and RMI

The services the Security Administrator is able to start and stop, and where this configuration is performed, is described as follows:

- TLS Server (RMI)
    - Configured via Local Console
- Syslog client service:
    - Configured via RMI
- IPsec service:
    - Configured via RMI

FMT_MOF.1/Services; FMT_SMF.1; FMT_SMR.2; FMT_MOF.1/ManualUpdate

7.4.2 Packet Filtering

While the TOE is powering up, the TSF network interfaces are disabled prior to completion of the power-up self-tests. This ensures that the TSF is operating properly and that the packet filtering rules have been initialized before the TSF allows the processing of network data on network interfaces. Once the network interfaces have been enabled, the TSF verifies that all packets the TOE receives are processed using the TSF firewall rulesets (rulesets can apply to inbound and/or outbound traffic). Any time that networking functionality is enabled, the firewall rules will be applied.

Packet filtering is designed to enforce network traffic flow policy at the IPsec GW Policy Enforcement Point (PEP) level. The TOE allows an authorized security administrator to configure packet filtering criteria, actions for a packet matching the criteria and whether packet logging is enabled. The network interfaces to this TSF are as follows:

- Black Network
    - Faces the external, public network
- Red Network
    - Faces the internal, private network
- Management Network
    - Private network separate from the Black and Red networks, used for administrative management of the TOE which includes access for syslog audit record transport over TLS, CRL download and HTTPS access to the RMI

The packet filtering criteria is based on the following criteria:

- IPv4 (RFC 791)
    - Source address
    - Destination Address
    - Protocol
- IPv6 (RFC 8200)
    - source address
    - destination address
    - next header (protocol)
- TCP (RFC 793)
    - Source Port
    - Destination Port
- UDP (RFC 768)
    - Source Port
    - Destination Port

The TSF supports the following packet processing actions:

- Allow
    - protect the packet with IPsec
- Drop
    - drop the packet with no further processing
- Bypass
    - allow the packet to flow through the TOE with no protection
- Log (can be combined with any other action listed above)
    - Generate audit record for the packet matching a rule

In addition to enforcing the packet processing rules above, the device enforces these rules in the order specified by the security administrator.

The packet processing rules are configured using the combination of device firewall rules and IPsec Security Policy Database (SPD) rules. Through the use of the RMI ‘IPsec Traffic Selectors’ REST API endpoint, a Security Administrator can configure these rules which will automatically be inserted into the TOE’s firewall configuration and IPsec SPD configuration. While the SA can modify the configured policies by adjusting FW rulesets from the CLI configuration management menu, the SA cannot fully configure the traffic selectors from the CLI alone. Note that any packet processing rules created using the ‘IPsec Traffic Selectors endpoint’ are only in effect while the IPsec service is running.

Note that the TOE automatically creates Firewall rules for the following services:

- Syslog client
    - allows for the establishment of a TLS session between the TSF and remote audit server
- RMI
    - allows for the establishment of an HTTPS session between the TSF and remote Security Administrator
- CDP
    - allows for the establishment of an HTTP session between the TSF and remote CRL distribution server to download any CRL necessary when performing certificate validation
        - When a DN to IP address mapping is configured, device firewall rules will also be put in place to allow communication to the configured IP addresses
        - Should X.509 certificates use CDP URIs that have IP addresses instead of DNs, firewall Allow rules must be configured instead of DNS mapping to allow the device to connect to CDPs and download a CRL

Packet processing rules that match ingress or egress network packets can generate a security audit log. This is configurable by a Security Administrator using the firewall rules. To enable or disable packet match event logging, a Security Administrator would use standard Linux iptables commands in conjunction with the CLI “Enter iptables command” menu option.

Packets not matching a rule are dropped by a hard-coded final rule.

The TSF implements support for IPv4, TCP, and UDP traffic. Correct implementation of these protocols has been established via third-party interoperability testing with known-good implementations of these common networking protocols.

The SA can configure packet filtering rules and apply those rules to any particular network interface. Note that the TSF's default policy of 'DROP' will always be applied to network packets that do not match explicitly-configured FW rules.

The TOE supports the processing of all IPv4 protocols listed in Table 3 of the [VPNSD], for packet filtering functionality. Only BYPASS (bypass is a permission of traffic through the firewall without passing through the IPsec processing), LOG and DROP rules are supported for IPv6 packet filtering. IPv6 is not supported for IPsec.

When the IPsec service is started, the Firewall (FW) rules required for basic IPsec operation are added to the iptables FW. When the service is stopped, such auto-generated FW rules are removed from the iptables FW. It must be recognized that when a security administrator adds rules to the FW manually and then performs IPsec service start/stop operations, the manually-entered rules will remain, but the IPsec service auto-generated rules will be removed from their current FW rule ordering and added to the end of the FW rule chain. The security administrator can use manual FW management described above to change the order using iptables "-D, --delete" and "-I, --insert" commands.

Figure 2 in Section 7.1.1 above, provides a block diagram of the major components that process network packets.

FPF_RUL_EXT.1

7.4.3 Management of TSF Data

Each administrative function identified in Section 7.4.1 above are accessible only through the RMI and CLI by a successfully identified and authenticated security administrator. None of these functions are accessible prior to security administrator log-in. The ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users by the access control functionality of the TOE operating system.

The ability to manage the TOE's trust store is restricted to authenticated security administrators by virtue of the TOE operating system's access control TSF.

FMT_MTD.1/CoreData

The cryptographic keys the Security Administrator is able to manage via the RMI and CLI include the following:

- Generate the following CSRs (which generates a public and private key pair) and import the subsequent signed CSR into the TOE for the following cases:
    - TOE's IPsec X.509 certificate/key_pair, to authenticate itself to a remote IPsec peer
    - TOE's TLS Server certificate/key_pair to authenticate itself to a remote TLS client for the RMI TSF
    - TOE's TLS Client certificate/key_pair to authenticate itself to the remote audit server
- Import of the following X.509 certificates:
    - Trusted Certificate Authority for use in authenticating the following end points:
        - Remote IPsec peer
        - Remote audit server
- Import CA(s) for the TOE's RMI TSF
- All cryptographic keys identified above are able to be deleted by the security administrator

Each trusted channel and trusted path TSF has its own certificate trust store, and thus, its own chain of trust, within the TOE.

FMT_MTD.1/CryptoKeys

7.5 Protection of the TSF

7.5.1 Protection of Administrator Passwords

Administrator passwords are stored salted and hashed using the SHA-512 hashing algorithm. No interface exists for the reading of the passwords stored on the TOE.

FPT_APW_EXT.1

7.5.2 TSF Testing

Upon bootup of the TOE, the TSF runs the following self-tests:

- Software power-on self testing (POST):
    - Known Answer Tests the Kernel, OpenSSL, and libgcrypt Cryptographic Modules, for each of the cryptographic algorithms utilized by the TSF
- Entropy Noise Source Health Check
    - Verification that RDSEED does not report a failure flag upon invocation of RDRAND by the TSF
- Digital Signature verification the TOE firmware

The TOE uses Intel® Secure Key technology (RDSEED). This technology provides a NIST SP 800-90 A, NIST SP 800-90 B and NIST SP 800-90 C compliant seed and RNG implementation. The entropy hardware in the 11th Gen Intel Core i5-1140 @ 2.6Ghz processor performs the required entropy noise tests. The Intel® Secure Key technology in the processor performs Built-In Self Tests (BISTs) to verify the correct operation of the Entropy Source (ES) prior to making the Digital Random Number Generator (DRNG) available to hosted software. The device verifies that the BIST was successful prior to using processor-provided DRNG data.

The TOE cryptographic module performs Known Answer Tests (KAT) on each algorithm implemented by the TSF. Each KAT consists of calling the algorithm with known inputs and verifying that the output matches the expected/”known” (pre-computed) output.

The TSF verifies the integrity of all TSF components by verifying a digital signature (ECDSA P-384) of the TOE firmware. The verification of the integrity of the TOE software ensures that the device software image matches the software image that was cryptographically authenticated and installed in the Virtual Machine (VM). This test involves using image digest checks to ensure the software image has not been altered, corrupt or otherwise modified.

Upon a failure of any of the self-tests above, the device provides a notification to a Security Administrator using the CLI and then shuts down. Because all cryptographic operations are tested, the TSF is known to be performing cryptographic operations correctly. Because the underlying hardware is tested, the TSF is known to be correctly executing the firmware. Together, when the TOE is operating, the TSF is known to be operating as expected to enforce the SFRs.

FPT_TST_EXT.1; FPT_FLS.1/SelfTest; FPT_TST_EXT.3

7.5.3 Trusted Update

The TOE provides means to query the current executing TOE software version via the CLI. The RMI provides the means to perform a trusted software update. The secure update process requires that the new software image is digitally signed as provided by Viasat. The candidate update package is obtained from Viasat via download from a customer portal webpage provided by Viasat.

Upon attempt to upgrade the TOE software, the digital signature is verified by the TSF automatically, and if the digital signature is valid, the upgrade is performed. An audit record is generated indicating success; otherwise, the upgrade is aborted, and an audit record is generated indicating a failure occurred. Customers are instructed to contact Viasat product support if the update attempt fails.

FPT_TUD_EXT.1

The certificates pertaining to the trusted update TSF are contained on the device in plaintext; however, their data are included as part of the digital signature that is verified during the power on self-tests. In addition, these certificates are not configurable/modifiable. These certificates are installed at the factory by Viasat. If these certificates are deemed invalid because the certificate has expired, the TSF will reject the update attempt.

Revocation status of these certificates are checked when the administrator attempts to update the TOE software.

FPT_TUD_EXT.2

7.5.4 Protection of TSF Data

The TSF does not utilize any pre-shared keys. The CLI and RMI do not provide an interface/command that allow for the reading of private key data. All keys are stored on the TOE in plaintext.

FPT_SKP_EXT.1

7.5.5 Reliable Time Stamps

The following TSF utilize the system time:

- Security Audit
  - Audit timestamps
- IPsec and TLS Trusted Channels
  - X.509 certificate validity date checks and CRL nextUpdate checks
- TOE’s certificate chain load
  - X.509 certificate validity date checks and CRL nextUpdate checks
- TOE software update digital signature verification
  - X.509 certificate validity date checks and CRL nextUpdate checks

The TSF supports two methods of keeping system time. The SA configures the TSF to utilize one of the available methods. The methods include the following:

- Hyper-V VM Time Synchronization
- Manual setting of the time via the CLI or RMI

When Hyper-V VM time synchronization feature is utilized, the Hyper-V Virtualization System will synchronize the TOE’s system clock with the Windows 10 host OS time. The time synchronization event occurs only when the TOE (VM) is paused then resumed in Hyper-V. Depending on the time difference between Windows OS system time and the TOE’s system time, reboot may or may not synchronize the VM to the Windows OS time. In all cases, the Security Administrator should use the CLI or the RMI as described in [AGD] to verify that the TOE’s time is set correctly.

FPT_STM_EXT.1

7.6 TOE Access

7.6.1 TSF-initiated Session Locking

The administrator can access the TSF via the CLI or RMI. The TSF displays a configurable advisory and consent message at the CLI and RMI interfaces.

FTA_SSL_EXT.1; FTA_TAB.1

7.6.2 TSF-initiated Termination

The CLI and RMI logon sessions are subject to an administrator-configurable inactivity timeout. They are configured independently through their respective interfaces. The CLI is configurable from 30 to 72,000 seconds and the RMI is configurable from 30 to 72,000 seconds. Upon reaching the timeout, the device terminates the login session and requires the Security Administrator to repeat the login process. This inactivity timeout value is configurable by the administrator.

FTA_SSL.3

7.6.3 User-initiated Termination

The administrator can terminate a CLI and RMI session by logging out. An active administrative session can be terminated upon request by the administrator. For the CLI, an Security Administrator enters the letter “Q” at the appropriate menu prompt, which terminates the session. For the RMI, an Security Administrator issues a delete request on the login token.

FTA_SSL.4

7.7 Trusted Path/Channels

7.7.1 Inter-TSF Trusted Channel

The TSF communicates with the following trusted IT entities in the OE:

- VPN Gateways via the IPsec protocol
    - TOE acts as an IPsec peer
    - Method of assured identification of the non-TSF endpoint is provided by validation of X.509 certificates
- Remote Syslog Servers via the TLSv1.2 protocol
    - TOE acts as a TLS Client supporting mutual authentication
    - Method of assured identification of the non-TSF endpoint is provided by validation of X.509 certificates

FTP_ITC.1; FTP_ITC.1/VPN

7.7.2 Trusted Path

The RMI is a REST API protected over HTTPS, that is accessed using a “REST API client tool” as described in Section 1.3.4 above. The ‘RMI REST API REFERENCE’ in Appendix A of the guidance documentation contains specific details on how to access this API. The RMI is the only means of remote administration of the TSF.

FTP_TRP.1/Admin

8. Terms and Definitions

Table 13: TOE Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
CA	Certificate Authority
CBC	Cipher Block Chaining
CLI	Command Line Interface
CMOS	Complementary Metal–Oxide–Semiconductor
CRL	Certificate Revocation List
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CTR	Counter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPs	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
KAT	Known Answer Test
KDF	Key Derivation Function
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standards
REST	Representational State Transfer
RFC	Requests for Comments
RMI	Remote Management Interface
RSA	Rivest-Shamir-Adleman
SA	Security Association
SAN	Subject Alternative Name
SFP	Small Form-Factor Pluggable
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SPD	Security Policy Database
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security

Table 13: TOE Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
UDP	User Datagram Protocol
UI	User Interface
URI	Uniform Resource Identifier
VPN	Virtual Private Network
vND	Virtual Network Device
WAN	Wide Area Network

Table 14: CC Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DOD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSF	TSF = TOE for pND or Case 1 vND according to section 1.2; TSF = TOE + VS for Case 2 vND (vND evaluated as a pND) according to section 1.2
TSFI	TSF Interface
TSS	TOE Summary Specification

9. References

Table 15: TOE Guidance Documentation

Reference	Description	Version	Date
[AGD]	Viasat Secure VPN User Guide Viasat Document No.: 1398812	Rev. 008	15 December 2023

Table 16: Common Criteria v3.1 References

Reference	Description	Version	Date
[C1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001	V3.1 R5	April 2017
[C2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2017-04-002	V3.1 R5	April 2017
[C3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2017-04-003	V3.1 R5	April 2017
[C4]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2017-04-004	V3.1 R5	April 2017
[C5]	CC and CEM addenda - Exact Conformance, Selection-Based SFRs, Optional SFRs CCDB-2017-05-xxx	V0.5	May 2017

Table 17: Supporting Documentation

Reference	Description	Version	Date
[cPP]	Collaborative Protection Profile for Network Devices	2.2e	March 23, 2020
[SD]	Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP	2.2	December 2019
[VPN]	PP-Module for Virtual Private Network (VPN) Gateways	1.2	2022-03-31
[VPNSD]	Supporting Document Mandatory Technical Document PP-Module for Virtual Private Network (VPN) Gateways	1.2	2022-03-31