Apriva MESA VPN v3.0 Security Target

Version 0.4
July 18, 2023

Prepared for:

Apriva ISS, LLC.
7600 N. 16th St, Suite 230
Phoenix, AZ 85020

Prepared By:

Gossamer
Laboratories
www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Apriva MESA VPN provided by Apriva ISS, LLC. The TOE is being evaluated as a VPN Gateway.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

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

1.1 Security Target Reference

ST Title – Apriva MESA VPN v3.0 Security Target

ST Version – Version 0.4

ST Date – July 18, 2023

1.2 TOE Reference

TOE Identification Apriva ISS, LLC Apriva MESA VPN 3.0

TOE Developer – Apriva ISS, LLC

Evaluation Sponsor – Apriva ISS, LLC

1.3 TOE Overview

The Target of Evaluation (TOE) is the Apriva MESA VPN v3.0. The Apriva MESA VPN server is an IPsec VPN gateway designed to provide mobile devices with a secure connection to a protected network. The Apriva MESA VPN is a standards-based VPN concentrator with no proprietary modes of operation, and supporting most native VPN clients.

1.4 TOE Description

The TOE is the Apriva MESA VPN 3.0 consisting of the following hardware and software.

- Dell PowerEdge R750 2U Rackmount Server or Dell PowerEdge R650 1U Rackmount Server
- CPU: Intel® Xeon® CPU listed below
- NICs:
  - Intel X710-T4L Quad Port 10GbE BASE-T Adapter
  - Broadcom 5720 Dual Port 1GbE BASE-T Adapter
- Running Apriva MESA VPN release 3.0

The TOE was tested using an Intel® Xeon® Silver 4310 (Ice Lake microarchitecture) processor. The following processor are all equivalent (same microarchitecture and instruction set) and may be used in the TOE.

- Intel® Xeon® Silver 4309Y
- Intel® Xeon® Silver 4310
- Intel® Xeon® Silver 4316
- Intel® Xeon® Silver 4314
- Intel® Xeon® Gold 5315Y
- Intel® Xeon® Gold 5317
- Intel® Xeon® Gold 5318Y
- Intel® Xeon® Gold 5318N
- Intel® Xeon® Gold 5320
- Intel® Xeon® Gold 6326
- Intel® Xeon® Gold 6330
- Intel® Xeon® Gold 6330N
- Intel® Xeon® Gold 6336Y
- Intel® Xeon® Gold 6338N

1.5 Physical Boundaries

Each TOE appliance runs the 3.0 version of the Apriva MESA VPN software and has physical network connections to its environment to facilitate managing and filtering network traffic. The TOE appliance can also be the destination of network traffic, where it provides interfaces for its own management.

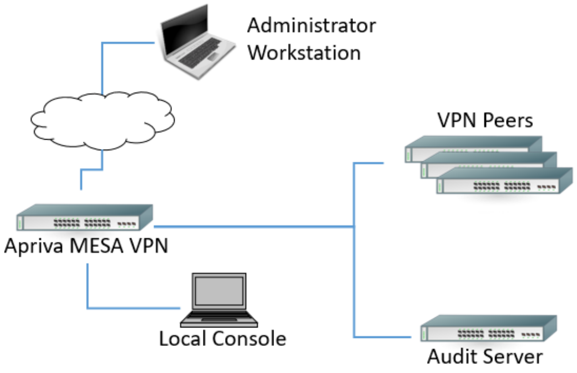

Figure 1-1 Generic TOE Environment

The TOE may be accessed and managed through a PC or terminal which can be remote from or directly connected to the TOE.

The TOE can be an IPsec peer or be a server for IPsec clients.

The TOE can be configured to forward its audit records to an external syslog server in the network environment. This is generally advisable given the limited audit log storage space on the evaluated appliances.

The TOE is delivered to the customer via courier. The Software is pre-installed on the hardware prior to delivery.

1.6 Logical Boundaries

This section summarizes the security functions provided by the Apriva MESA VPN:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

1.6.1 Security audit

The TOE is able to generate logs for a wide range of security relevant events. The TOE always stores the logs locally so they can be accessed by an administrator and can be configured to send the logs to a designated log server using TLS to protect the logs while in transit on the network.

1.6.2 Cryptographic support

The TOE contains a CAVP-tested cryptographic module that provides key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher-level cryptographic protocols including IPsec, TLS, and SSH.

1.6.3 Identification and authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of specific ICMP response. It provides the ability to perform password and public key authentication for administrative users.

1.6.4 Security management

The TOE implements a limited command line interface (CLI) to allow authorized administrators to configure the TOE. This interface restricts the administrator to executing commands required to configure and administer the TOE. All administrative activity and functions including security management commands are limited to authorized users (i.e., administrators) only after they have provided acceptable user identification and authentication data to the TOE.

1.6.5 Packet Filtering

The TOE provides extensive packet filtering capabilities for IPv4, IPv6, TCP, and UDP. The authorized administrator can define packet filtering rules that apply to most every field within the identified packet types. The authorized administrator can define each rule to permit, deny, and log each decision.

1.6.6 Protection of the TSF

The TOE implements a number of features to protect itself to ensure the reliability and integrity of its security features.

It protects particularly sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. It also provides its own timing mechanism to ensure that reliable time information is available (e.g., for log accountability).

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms (i.e., verification of the digital signature of each new image) so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

1.6.7 TOE access

The TOE can be configured to display a login banner when an administrator establishes an interactive session and subsequently will enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated. The TOE can also restrict VPN clients based on location and time and can assign a private VPN address to a client.

1.6.8 Trusted path/channels

The TOE protects interactive communication with administrators using SSH for CLI access to ensure both integrity and disclosure protection. If the negotiation of an encrypted session fails or if the user does not have authorization for remote administration, an attempted connection will not be established.

The TOE protects communication with the audit log server using TLS connections to prevent unintended disclosure or modification of logs. The TOE can establish IPsec connections with clients and peers.

1.7 TOE Documentation

Apriva offers a series of documents that describe the installation of the TOE as well as guidance for subsequent use and administration of the applicable security features of the VPN Gateway. The following document was examined as part of the evaluation:

- Apriva MESA VPN Common Criteria Guidance

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
  - Part 3 Conformant
- Package Claims:
  - PP-Configuration for Network Device and Virtual Private Network (VPN) Gateways, version 1.2, 31 March 2022
    - Base-PP: 'collaborative Protection Profile for Network Devices', Version 2.2e, 23 March 2020/
    - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways', 1.2, 31 March 2022 (NDcPP22e/VPNGW12)

Package	Technical Decision	Applied	Notes
CPP_ND_V2.2E	TD0738 – NIT Technical Decision for Link to Allowed-With List	Yes	
CPP_ND_V2.2E	TD0670 – NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Yes	
CPP_ND_V2.2E	TD0639 - NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
CPP_ND_V2.2E	TD0638 - NIT Technical Decision for Key Pair Generation for Authentication	Yes	
CPP_ND_V2.2E	TD0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	SFR not claimed
CPP_ND_V2.2E	TD0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters	No	SFR not claimed
CPP_ND_V2.2E	TD0634 - NIT Technical Decision for Clarification required for testing IPv6	Yes	
CPP_ND_V2.2E	TD0633 - NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
CPP_ND_V2.2E	TD0632 - NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
CPP_ND_V2.2E	TD0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
CPP_ND_V2.2E	TD0592 - NIT Technical Decision for Local Storage of Audit Records	Yes	
CPP_ND_V2.2E	TD0591 - NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
CPP_ND_V2.2E	TD0581 - NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
CPP_ND_V2.2E	TD0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
CPP_ND_V2.2E	TD0572 - NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
CPP_ND_V2.2E	TD0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0570 - NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	SFR not claimed
CPP_ND_V2.2E	TD0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	

CPP_ND_V2.2E	TD0563 - NiT Technical Decision for Clarification of audit date information	Yes	
CPP_ND_V2.2E	TD0556 - NIT Technical Decision for RFC 5077 question	No	SFR not claimed
CPP_ND_V2.2E	TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	SFR not claimed
CPP_ND_V2.2E	TD0547 - NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
CPP_ND_V2.2E	TD0546 - NIT Technical Decision for DTLS - clarification of Application Note 63	No	SFR not claimed
CPP_ND_V2.2E	TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
CPP_ND_V2.2E	TD0536 - NIT Technical Decision for Update Verification Inconsistency	Yes	
CPP_ND_V2.2E	TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	SFR not claimed
CPP_ND_V2.2E	TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
MOD_VPNGW_v1.2	TD0723 – Correction to ECDSA Curve Selection	Yes	
MOD_VPNGW_v1.2	TD0683: RFC 2460 to be replaced with RFC 8200	Yes	
MOD_VPNGW_v1.2	TD0657: IPSEC_EXT.1.6 GCM support for VPN GW	Yes	
MOD_VPNGW_v1.2	TD0656: Missing EAs for VPN GW Optional Headend SFRs	No	SFR not claimed

2.1 Conformance Rationale

The ST conforms to the NDcPP22e/VPNGW12. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the NDcPP22e/VPNGW12 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP22e/VPNGW12 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP22e/VPNGW12 should be consulted if there is interest in that material.

In general, the NDcPP22e/VPNGW12 has defined Security Objectives appropriate for VPN Gateways and as such are applicable to the Apriva MESA VPN TOE.

3.1 Security Objectives for the Operational Environment

OE.ADMIN_CREDENTIALS_SECURE The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

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

OE.CONNECTIONS See TD0520 for SARs.
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.

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

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

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

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

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.
For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.

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

OE.VM_CONFIGURATION (applies to vNDs only)
For vNDs, the Security Administrator ensures that the VS and VMs are configured to
- reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and
- correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualisation features such as cloning, save/restore, suspend/resume, and live migration.

If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP22e/VPNGW12. The NDcPP22e/VPNGW12 defines the following extended requirements and since they are not redefined in this ST the NDcPP22e/VPNGW12 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0657
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
- NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0634
- NDcPP22e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
- NDcPP22e:FIA_PMG_EXT.1: Password Management
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- VPNGW12:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e/VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW12:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW12:FPF_RUL_EXT.1: Packet Filtering Rules
- NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
- NDcPP22e/VPNGW12:FPT_TST_EXT.1: TSF Testing
- VPNGW12:FPT_TST_EXT.3: Self-Test with Defined Methods
- NDcPP22e/VPNGW12:FPT_TUD_EXT.1: Trusted Update
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
- VPNGW12:FTA_VCM_EXT.1: VPN Client Management - per TD0656

5. Security Requirements

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

The SFRs have all been drawn from the NDcPP22e/VPNGW12. The refinements and operations already performed in the NDcPP22e/VPNGW12 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP22e/VPNGW12 and any residual operations have been completed herein. Of particular note, the NDcPP22e/VPNGW12 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP22e/VPNGW12. The NDcPP22e/VPNGW12 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Apriva MESA VPN TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP22e:FAU_GEN.1: Audit Data Generation
	VPNGW12:FAU_GEN.1/VPN: Audit Data Generation (VPN Gateway)
	NDcPP22e:FAU_GEN.2: User identity association
	NDcPP22e:FAU_STG.1: Protected audit trail storage
	NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP22e:FCS_CKM.1: Cryptographic Key Generation
	VPNGW12:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication)
	NDcPP22e:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP22e:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP22e/VPNGW12:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP22e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP22e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP22e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0657
	NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
	NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0634
	NDcPP22e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
FIA: Identification and authentication	NDcPP22e:FIA_AFL.1: Authentication Failure Management
	NDcPP22e:FIA_PMG_EXT.1: Password Management
	NDcPP22e:FIA_UAU.7: Protected Authentication Feedback
	NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	VPNGW12:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP22e/VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	VPNGW12:FMT_MTD.1/CryptoKeys: Management of TSF Data

	NDcPP22e:FMT_SMF.1: Specification of Management Functions - per TD0631
	VPNGW12:FMT_SMF.1/VPN: Specification of Management Functions
	NDcPP22e:FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	VPNGW12:FPF_RUL_EXT.1: Packet Filtering Rules
FPT: Protection of the TSF	NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
	VPNGW12:FPT_FLS.1/SelfTest: Failure with Preservation of Secure State (Self-Test Failures)
	NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
	NDcPP22e/VPNGW12:FPT_TST_EXT.1: TSF Testing
	VPNGW12:FPT_TST_EXT.3: Self-Test with Defined Methods
	NDcPP22e/VPNGW12:FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	NDcPP22e:FTA_SSL.3: TSF-initiated Termination
	NDcPP22e:FTA_SSL.4: User-initiated Termination
	NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP22e:FTA_TAB.1: Default TOE Access Banners
	VPNGW12:FTA_TSE.1: TOE Session Establishment - per TD0656
	VPNGW12:FTA_VCM_EXT.1: VPN Client Management - per TD0656
FTP: Trusted path/channels	NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel - per TD0639
	VPNGW12:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications)
	NDcPP22e:FTP_TRP.1/Admin: Trusted Path - per TD0639

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP22e/VPNGW12:FAU_GEN.1)

NDcPP22e/VPNGW12: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**.

Requirement	Auditable Events	Additional Content
NDcPP22e/VPNGW12:FAU_GEN.1	None	None
NDcPP22e:FAU_GEN.2	None	None
NDcPP22e:FAU_STG.1	None	None
NDcPP22e:FAU_STG_EXT.1	None	None
NDcPP22e:FCS_CKM.1	None	None
VPNGW12:FCS_CKM.1/IKE	None	None
NDcPP22e:FCS_CKM.2	None	None
NDcPP22e:FCS_CKM.4	None	None
NDcPP22e/VPNGW12:FCS_COP.1/DataEncryption	None	None

NDcPP22e:FCS_COP.1/Hash	None	None
NDcPP22e:FCS_COP.1/KeyedHash	None	None
NDcPP22e:FCS_COP.1/SigGen	None	None
NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA Session Establishment with peer	Reason for failure Entire packet contents of packets transmitted/received during session establishment
NDcPP22e:FCS_RBG_EXT.1	None	None
NDcPP22e:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP22e:FCS_TLSC_EXT.1	Failure to establish a TLS Session.	Reason for failure.
NDcPP22e:FCS_TLSC_EXT.2	None	None
NDcPP22e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_PMG_EXT.1	None	None
NDcPP22e:FIA_UAU.7	None	None
NDcPP22e:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
NDcPP22e/VPNGW12:FIA_X509_EXT.2	None	None
NDcPP22e:FIA_X509_EXT.3	None	None
NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP22e:FMT_MTD.1/CoreData	None	None
NDcPP22e:FMT_MTD.1/CryptoKeys	None	None
VPNGW12:FMT_MTD.1/CryptoKeys	None	None
NDcPP22e:FMT_SMF.1	All management activities of TSF data.	None
VPNGW12:FMT_SMF.1/VPN	None	None
NDcPP22e:FMT_SMR.2	None	None
VPNGW12:FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol
NDcPP22e:FPT_APW_EXT.1	None	None
VPNGW12:FPT_FLS.1/SelfTest	None	None
NDcPP22e:FPT_SKP_EXT.1	None	None
NDcPP22e:FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
NDcPP22e/VPNGW12:FPT_TST_EXT.1	None	None
VPNGW12:FPT_TST_EXT.3	None	None
NDcPP22e/VPNGW12:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None

NDcPP22e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
NDcPP22e:FTA_SSL.4	The termination of an interactive session.	None
NDcPP22e:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	None
NDcPP22e:FTA_TAB.1	None	None
VPNGW12:FTA_TSE.1	None	None
VPNGW12:FTA_VCM_EXT.1	None	None
NDcPP22e:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
VPNGW12:FTP_ITC.1/VPN	None	None
NDcPP22e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None

Table 2 Auditable Events

NDcPP22e/VPNGW12: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**.

5.1.1.2 User identity association (NDcPP22e:FAU_GEN.2)

NDcPP22e:FAU_GEN.2.1

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

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

NDcPP22e:FAU_STG.1.1

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

NDcPP22e:FAU_STG.1.2

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

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

NDcPP22e:FAU_STG_EXT.1.1

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

NDcPP22e:FAU_STG_EXT.1.2

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

NDcPP22e:FAU_STG_EXT.1.3

The TSF shall [***overwrite previous audit records according to the following rule: [perform a log rotation]***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

NDcPP22e:FCS_CKM.1.1

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

5.1.2.2 Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW12:FCS_CKM.1/IKE)

VPNGW12:FCS_CKM.1.1/IKE

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

- ***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes,***
- ***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-384 and [P-256, P-521]***

] and [

- ***no other key generation algorithm***

] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. (TD0723 applied)

5.1.2.3 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

NDcPP22e:FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, 'Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1,***
- ***Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' (TD0581 applied),***
- ***FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [groups listed in RFC 3526] (TD0580 applied)***

].

5.1.2.4 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

NDcPP22e:FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***logically addresses the storage location of the key and performs a [[2]-pass] overwrite consisting of [a pseudo-random pattern using the TSF's RBG, zeroes]***]

that meets the following: No Standard.

5.1.2.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e/VPNGW12:FCS_COP.1/DataEncryption)

NDcPP22e/VPNGW12:FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] and [***CTR***] mode and cryptographic key sizes [***128 bits, 256 bits***] , and [***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***].

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

NDcPP22e:FCS_COP.1.1/Hash

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

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

NDcPP22e:FCS_COP.1.1/KeyedHash

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

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

NDcPP22e:FCS_COP.1.1/SigGen

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

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

that meet the following: [

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

].

5.1.2.9 IPsec Protocol - per TD0657 (NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1)

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.1

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

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.2

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

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.3

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

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)***] and [***no other algorithm***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.5

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

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 5282)***].

NDcPP22e/VPNGW12: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 [1/12 to 24] hours]***].

NDcPP22e/VPNGW12: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 [1/12 to 24] hours]***].

NDcPP22e/VPNGW12: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 [**275 (groups 14 and 24), 320 (group 19), 325(group 15), 480(group 20)**] bits.

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv2***] exchanges of length [***at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s)
- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and
- [***[14 (2048-bit MODP), 15 (3072-bit MODP)] according to RFC 3526, [24 (2048-bit MODP with 256-bit POS)] according to RFC 5114***].

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.12

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

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.13

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

NDcPP22e/VPNGW12: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***].

5.1.2.10 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

NDcPP22e:FCS_RBG_EXT.1.1

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

NDcPP22e:FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[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 18031:2011Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

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

NDcPP22e:FCS_SSHS_EXT.1.1

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

NDcPP22e:FCS_SSHS_EXT.1.2

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

NDcPP22e:FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**262127**] bytes in an SSH transport connection are dropped.

NDcPP22e:FCS_SSHS_EXT.1.4

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

NDcPP22e:FCS_SSHS_EXT.1.5

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

NDcPP22e:FCS_SSHS_EXT.1.6

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

NDcPP22e:FCS_SSHS_EXT.1.7

The TSF shall ensure that [***ecdh-sha2-nistp256***] and [***ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

NDcPP22e:FCS_SSHS_EXT.1.8

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

5.1.2.12 TLS Client Protocol Without Mutual Authentication - per TD0634 (NDcPP22e:FCS_TLSC_EXT.1)

NDcPP22e: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,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]

and no other ciphersuites.

NDcPP22e:FCS_TLSC_EXT.1.2

The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 section 6, IPv4 address in SAN***].

NDcPP22e: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 [***require administrator authorization to establish the connection if the TSF fails to [determine the revocation status] of the presented server certificate***].

NDcPP22e: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/groups***] in the Client Hello.

5.1.2.13 TLS Client Support for Mutual Authentication (NDcPP22e:FCS_TLSC_EXT.2)

NDcPP22e:FCS_TLSC_EXT.2.1

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

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

NDcPP22e:FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within [**2-10**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

NDcPP22e:FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [the unlock action] is taken by an Administrator***].

5.1.3.2 Password Management (NDcPP22e:FIA_PMG_EXT.1)

NDcPP22e:FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [

'!', '@', '#', '$', '%', '^', '&', '*', '(', ')',
[''', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?' , '[', '_']', '_', '`', '', '|', '~', '<space>']

];

b) Minimum password length shall be configurable to between [**8**] and [**64**] characters.

5.1.3.3 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

NDcPP22e:FIA_UAU.7.1

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

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

NDcPP22e:FIA_UAU_EXT.2.1

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

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

NDcPP22e:FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- [***[Respond to ICMP Echo Request with an Echo Reply,***
- ***Respond with ICMP Destination Unreachable messages]***].

NDcPP22e:FIA_UIA_EXT.1.2

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

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

NDcPP22e:FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960***]
- 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.

NDcPP22e:FIA_X509_EXT.1.2/Rev

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

5.1.3.7 X.509 Certificate Validation (VPNGW12:FIA_X509_EXT.1/Rev)

VPNGW12:FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***]
- 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.

VPNGW12:FIA_X509_EXT.1.2/Rev

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

5.1.3.8 X.509 Certificate Authentication (NDcPP22e/VPNGW12:FIA_X509_EXT.2)

NDcPP22e/VPNGW12:FIA_X509_EXT.2.1

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

NDcPP22e/VPNGW12:FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall (choose one of:) [***allow the Administrator to choose whether to accept the certificate in these cases***].

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

NDcPP22e:FIA_X509_EXT.3.1

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

NDcPP22e:FIA_X509_EXT.3.2

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

5.1.4 Security management (FMT)

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

NDcPP22e:FMT_MOF.1.1/ManualUpdate

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

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

NDcPP22e:FMT_MTD.1.1/CoreData

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

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

NDcPP22e:FMT_MTD.1.1/CryptoKeys

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

5.1.4.4 Management of TSF Data (VPNGW12:FMT_MTD.1/CryptoKeys)

VPNGW12:FMT_MTD.1.1/CryptoKeys

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

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

NDcPP22e:FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to start and stop services,***
- ***Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),***
- ***Ability to modify the behavior of the transmission of audit data to an external IT entity,***
- ***Ability to manage the cryptographic keys,***
- ***Ability to configure the lifetime for IPsec SAs,***
- ***Ability to re-enable an Administrator account,***
- ***Ability to set the time which is used for time-stamps,***
- ***Ability to configure the reference identifier for the peer,***

- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to import X509v3 certificates to the TOE's trust store,
- Ability to manage the trusted public keys database].

5.1.4.6 Specification of Management Functions (VPNGW12:FMT_SMF.1/VPN)

VPNGW12:FMT_SMF.1.1/VPN

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

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

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

NDcPP22e:FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

NDcPP22e:FMT_SMR.2.2

The TSF shall be able to associate users with roles.

NDcPP22e:FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

5.1.5 Packet Filtering (FPF)

5.1.5.1 Packet Filtering Rules (VPNGW12:FPF_RUL_EXT.1)

VPNGW12:FPF_RUL_EXT.1.1

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

VPNGW12:FPF_RUL_EXT.1.2

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

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

VPNGW12: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.

VPNGW12:FPF_RUL_EXT.1.4

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

VPNGW12:FPF_RUL_EXT.1.5

The TSF shall process the applicable Packet Filtering rules (as determined in accordance with VPNGW12:FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

VPNGW12:FPF_RUL_EXT.1.6

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

5.1.6 Protection of the TSF (FPT)

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

NDcPP22e:FPT_APW_EXT.1.1

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

NDcPP22e:FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

5.1.6.2 Failure with Preservation of Secure State (Self-Test Failures) (VPNGW12:FPT_FLS.1/SelfTest)

VPNGW12:FPT_FLS.1.1/SelfTest

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

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

NDcPP22e:FPT_SKP_EXT.1.1

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

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

NDcPP22e:FPT_STM_EXT.1.1

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

NDcPP22e:FPT_STM_EXT.1.2

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

5.1.6.5 TSF Testing (NDcPP22e/VPNGW12:FPT_TST_EXT.1)

NDcPP22e/VPNGW12:FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [***during initial startup (on power on), at the conditions [when a crypto module is loaded into memory, when reading entropy, when a software component is loaded into memory, when a configuration file is read]***] to demonstrate the correct operation of the TSF: noise source health tests, [

- ***Power-On Self Test (POST)***
  - ***FIPS Self-Test for the Kernel, OpenSSL, and libgcrypt Cryptographic Modules***
    - ***HMAC Integrity Check***
    - ***Known Answer Tests of cryptographic algorithms***
  - ***Entropy Health Check***
    - ***Verification that RDSEED does not report a failure***
  - ***Digital Signature verification of all software components***
- ***Loading a Crypto Module***
  - ***FIPS Self-Test***
    - ***Integrity Check***
    - ***Known Answer Tests of cryptographic algorithms***
- ***Reading Entropy***
  - ***Verification that RDSEED does not report a failure***
- ***Loading a Software Component***
  - ***Digital Signature verification***
- ***Reading a Configuration File***
  - ***Hash Verification (comparison with the hash generated when the configuration was updated).]***

5.1.6.6 Self-Test with Defined Methods (VPNGW12:FPT_TST_EXT.3)

VPNGW12: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.

VPNGW12:FPT_TST_EXT.3.2

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

5.1.6.7 Trusted Update (NDcPP22e/VPNGW12:FPT_TUD_EXT.1)

NDcPP22e/VPNGW12:FPT_TUD_EXT.1.1

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

NDcPP22e/VPNGW12:FPT_TUD_EXT.1.2

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

NDcPP22e/VPNGW12:FPT_TUD_EXT.1.3

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

5.1.7 TOE access (FTA)

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

NDcPP22e:FTA_SSL.3.1

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

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

NDcPP22e:FTA_SSL.4.1

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

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

NDcPP22e:FTA_SSL_EXT.1.1

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

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

NDcPP22e:FTA_TAB.1.1

Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.7.5 TOE Session Establishment - per TD0656 (VPNGW12:FTA_TSE.1)

VPNGW12:FTA_TSE.1.1

The TSF shall be able to deny session establishment of a remote VPN client session based on location, time, day, [***no other attributes***].

5.1.7.6 VPN Client Management - per TD0656 (VPNGW12:FTA_VCM_EXT.1)

VPNGW12:FTA_VCM_EXT.1.1

The TSF shall assign a private IP address to a VPN client upon successful establishment of a security session.

5.1.8 Trusted path/channels (FTP)

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

NDcPP22e:FTP_ITC.1.1

The TSF shall be capable of using [***IPsec, TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server ***(TLS)***, [***[VPN communications (IPsec)]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

NDcPP22e:FTP_ITC.1.2

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

NDcPP22e:FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [**audit server, VPN communications**].

5.1.8.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW12:FTP_ITC.1/VPN)

VPNGW12: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.

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

NDcPP22e:FTP_TRP.1.1/Admin

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

NDcPP22e:FTP_TRP.1.2/Admin

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

NDcPP22e:FTP_TRP.1.3/Admin

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

5.2 TOE Security Assurance Requirements

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

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

Table 3 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

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

ADV_FSP.1.1c

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

ADV_FSP.1.2c

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

ADV_FSP.1.3c

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

ADV_FSP.1.4c

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

ADV_FSP.1.1e

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

ADV_FSP.1.2e

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

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

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

AGD_OPE.1.2c

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

AGD_OPE.1.3c

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

AGD_OPE.1.4c

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

AGD_OPE.1.5c

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

AGD_OPE.1.6c

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

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d

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

AGD_PRE.1.1c

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

AGD_PRE.1.2c

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

AGD_PRE.1.1e

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

AGD_PRE.1.2e

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

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

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

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

ATE_IND.1.1d

The developer shall provide the TOE for testing.

ATE_IND.1.1c

The TOE shall be suitable for testing.

ATE_IND.1.1e

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

ATE_IND.1.2e

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

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

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

AVA_VAN.1.3e

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

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TSF generates and formats audit logs according to RFC 5424 and include the Pri, Version, Timestamp, Hostname, App-Name, and Msg fields. The Structured-Data, ProcID, and MsgID fields contain NILVALUE. The Pri, Version, and Hostname fields are not relevant to Common Criteria, but may be used to filter audit records once they have been transmitted from the TSF. The Timestamp field specifies the date/time the audit log was generated down to the nearest second. The App-Name field contains the name of the process that generated the audit log. When there is not an external user associated with the audit event, the App-Name field is the subject identity. The TSF uses the Msg field to fulfill the remaining audit requirements. For user generated audit events, the Msg field includes the user's username or X.509 Distinguished Name. The Msg field includes text that describes the audit event (type of event and success or failure) and includes any additional details listed in **Table 2**. The TSF generates the audit records for startup and shutdown of the audit function, the administrative actions described in Section 6.4, and the events listed in **Table 2**.

Keys affected by cryptographic operations (generation, deletion) are identified in logs using a generic identifier. Because the TOE can only have one key of each kind, they are identified by their purpose (e.g., "VPN-certificatename", "TLS-certificatename", "SSH-username-keyname", etc.)

The TSF generates the audit records for each packet filter LOG rule that is configured. These audit records include the network interface, source IP address, destination IP address, transport layer protocol, source port, and destination port. If a network interface of the TSF receives network traffic faster than it can process it, it drops traffic when its receive queue grows beyond 256 outstanding packets. The TSF ensures it can always log dropped packets by aggregating multiple dropped packets into a single log; it logs the number packets that have been dropped once per minute.

The TOE is a single standalone device that stores audit data locally. The TSF functions as an Originator and transmits audit logs to a Collector (syslog server) using Syslog over TCP as specified in RFCs 5424, 5425, and 6587. RFC 5424 specifies how the TSF formats logs for local storage and for transmission to the syslog server. The TSF sends audit records to the syslog server simultaneously with the local logging operation. The TSF uses TLS to secure the connection with the syslog server. If the link to the syslog server is down and cannot be established, the TSF will continue to store audit records locally. When the syslog server becomes operational, the TSF resumes transmitting new audit logs to the server. The logs generated while the syslog server was unavailable are not transmitted to the syslog server.

The TSF stores local logs in /var/log. The Pri and Version fields are not recorded in the local log files. The TSF stores the audit logs it generates as the following discrete local log types: iptables, secure, messages, quicksec, syslog, diag, boot, sysman, common. To prevent unauthorized access, modification, or deletion of the logs the TSF command line interface does not provide functions for users to directly access the /var/log directory. The command line interface allows authorized users to view logs via a "show log" command, which provides the user with a read only interface to the log files. Unauthorized users are prevented from accessing the CLI of the TOE, which prevents them from executing any commands for deletion of audit files or viewing of the contents of audit files. All log types are protected local and securely forwarded to an external syslog server in the same manner.

/var/log is a dedicated 10GB partition for local audit log storage. It is unlikely that the local audit storage will become full, because for each log type, the logging subsystem of the TSF performs log rotation. The log rotation is based on time or file size, whichever comes first. Time based rotation occurs daily at an administrator-configured time. File size rotation occurs when the log files reach an administrator-configured size from 1MB to 1000MB, default 100MB. Each local log type is rotated independently of the other local log types. When rotating logs, the TSF compresses the active log file, creates a new log file, and checks to see if the maximum number of archives has been exceeded. If the maximum number of archives has been exceeded, the TSF deletes the oldest archive. The TSF keeps a default of seven archives. The TSF also performs a log rotate on all logs if the audit storage reaches 90% of capacity.

The Security audit function satisfies the following security functional requirements:

- NDcPP22e/VPNGW12:FAU_GEN.1/VPN: The TOE generates all the required audit events in **Table 2**. Each audit record identifies the date/time, event type, outcome of the event, responsible subject/user, as well as the additional event-specific content indicated in **Table 2**. In addition, audits for generating, importing, changing, or deleting cryptographic keys identify the key using the configured name.
- NDcPP22e:FAU_GEN.2: The TOE identifies the responsible user for each event based on the specific administrator or network entity (identified by IP address) that caused the event.
- NDcPP22e:FAU_STG.1: For all log types, the TOE restricts direct access to the audit log.
- NDcPP22e:FAU_STG_EXT.1: The TOE can be configured to export audit records from all log types to an external SYSLOG server. This communication is protected with the use of TLS.

6.2 Cryptographic support

The TOE supports a range of cryptographic services using the following cryptographic libraries. The following functions have been CAVP tested. The TOE runs Red Hat Enterprise Linux v7.9 on the Intel® Xeon® Silver 4310 (Ice Lake) and uses the following cryptographic libraries to perform the cryptographic operations.

- Apriva ISS OpenSSL implementation
    - TLS connection, SSH connection, Key generation and establishment,
    - IPsec IKE cryptography,
    - Random number generator
- Apriva ISS Kernel Crypto API implementation
    - IPsec ESP data authentication and encryption
    - IPsec ESP HMAC
- Apriva ISS libgcrypt implementation
    - Trusted updates, Product integrity

Functions	Requirement	Standard	Certificate #
Encryption/Decryption			
AES CBC, CTR (128 and 256 bits)	FCS_COP.1/DataEncryption	FIPS Pub 197 ISO 10116 NIST SP 800-38A ISO 19772	A3908
AES GCM (128 and 256 bits)			A3913
Cryptographic hashing			
SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	FIPS Pub 180-4 ISO/IEC 10118-3:2004	A3921
Keyed-hash message authentication			
HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	FCS_COP.1/KeyedHash	FIPS Pub 198-1	A3921
Cryptographic signature services			
RSA Digital Signature Algorithm (rDSA) (2048, 3072 bits)	FCS_COP.1/SigGen	FIPS Pub 186-4 ISO/IEC 9796-2	A3921
Elliptic Curve Digital Signature Algorithm (P-256, P-384, P-521)	FCS_COP.1/SigGen	FIPS Pub 186-4 ISO/IEC 14888-3	A3921
Random bit generation			
CTR_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism	FCS_RBG_EXT.1	FIPS SP 800-90A ISO/IEC 18031:2011	A3908
Key generation			
RSA Key Generation (2048, 3072 bits)	FCS_CKM.1 FCS_CKM.1/IKE	FIPS Pub 186-4	A3921
ECC Key Generation (P-256, P-384, P-521)	FCS_CKM.1 FCS_CKM.1/IKE	FIPS Pub 186-4	A3921

Key establishment			
RSA	FCS_CKM.2	RSAES-PKCS1-v1_5	Verification by known good impl
KAS ECC	FCS_CKM.2	NIST SP 800-56A Rev 3	A3921
FFC Schemes using safe-prime groups	FCS_CKM.2	NIST SP 800-56A Rev 3	Verification by known good impl.

Table 4 Apriva ISS OpenSSL CAVP Certificates

Functions	Requirement	Standard	Certificate #
Encryption/Decryption			
AES CBC, GCM (128 and 256 bits)	FCS_COP.1/DataEncryption	FIPS Pub 197 ISO 10116 NIST SP 800-38A ISO 19772	A3903
Cryptographic hashing			
SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	FIPS Pub 180-4 ISO/IEC 10118-3:2004	A3907
Keyed-hash message authentication			
HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	FCS_COP.1/KeyedHash	FIPS Pub 198-1	A3907

Table 5 Apriva ISS Kernel Crypto API CAVP Certificates

Functions	Requirement	Standard	Certificate #
Cryptographic signature services			
RSA Digital Signature Algorithm (rDSA) (2048 bits)	FCS_COP.1/SigGen	FIPS Pub 186-4 ISO/IEC 9796-2	A3911
Cryptographic hashing			
SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	FIPS Pub 180-4 ISO/IEC 10118-3:2004	A3911
Keyed-hash message authentication			
HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	FCS_COP.1/KeyedHash	FIPS Pub 198-1	A3911

Table 6 Apriva ISS libgcrypt CAVP Certificates

The DRBG instantiated by the TSF is seeded with an estimated 256-bits of entropy. The entropy is gathered from the third party RDSEED entropy source provided by the CPU.

IPsec Protocol

The TSF implements IPsec as specified in RFCs 3602, 4301, 4303, 4106, 4868, 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 6.5. Packets not processed by any rules are dropped by a hard-coded final rule that may also log the dropped packet, if configured. Logging of dropped packets is desirable in situations where a network device is expected to filter packets ahead of the VPN. A dropped packet on the VPN gives an indication that the packet filtering configuration may be incorrect.

The TSF uses the QuickSec library to implement the IPsec protocol and performs all of its cryptographic operations using the CAVP tested OpenSSL library and kernel crypto.

The TSF can be configured to use the AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES- CBC-256 algorithms from the Red Hat Kernel Cryptographic Module for encryption and message authentication for IPsec ESP. When AES-CBC is negotiated as the symmetric cipher, the TOE supports HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

The TSF uses AES-CBC-128, AES-CBC-256, AES-GCM-128 or AES-GCM-256 to encrypt the IKEv2 payload and HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512 to authenticate the IKEv2 payload.

The TSF supports IKE_SA and Child_SA lifetime configurations. Either lifetime can be configured from 5 minutes to 24 hours. The IKE_SA has a default value of 24 hours, while the Child_SA has a default value of 8 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 Child_SA lifetime like acts as an idle timeout value. If any packets were processed through the Child_SA tunnel, the tunnel is re-keyed instead. The administrator can also configure the Child_SA tunnel to always rekey rather than drop the tunnel when no packets are processed.

The TSF supports DH groups 14, 15, 19, 20 and 24 for use in IKEv2. One or more of these groups may be enabled by the administrator. The TSF will negotiate the algorithms in the following order if multiple are enabled: 20, 19, 15, 24, and 14. The TSF generates the ephemeral private key (x) sizes used in Diffie-Hellman based on the negotiated group. This indicates that the TOE generates keys with FFC schemes using safe-prime groups that meet NIST SP 800-56A Revision 3 and per RFC3526/5114 for DH14, DH15, DH19, DH20 and DH24.

The TSF negotiates the allowed groups with the client in the IKEv2 exchange. The TSF will not allow the client to use a group that was not previously configured by the administrator. For example, if the client has selected group 5, the TSF will refuse to connect.

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 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 security management interface ensures that the Key Size(s) configurable for a CHILD_SA are less than or equal to the Key Size(s) configured for the IKEv2 SA. If a client attempts to negotiate a CHILD_SA with a key size that has not been configured on the TSF, the connection 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 Apriva MESA VPN supports authenticating a device connection using the Distinguished Name (DN) field of an X509 certificate. The DN contained in an x509 certificate presented during a VPN session negotiation must match the DN field types configured as required by the TOE (e.g., O, OU, C). Also, the common name (CN) field within the DN must be present in the list of approved device ids configured on the TOE. Only if the certificate's CN matches an approved device, and all DN field types match the TOE configuration, will the TOE accept the IPsec connection.

The TSF can generate keys in creation of a Certificate Signing Request (CSR). These keys can be used to allow the TSF to authenticate itself to peers using X.509 certificates with any of the following algorithm/keysize combinations:

- RSA 2048-bit
- RSA 3072-bit
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

SSH Server Protocol

The TSF implements SSHv2 according to RFCs 4251, 4252, 4253, 4254, 4344, 5656, and 6668. No options included in the RFCs have been implemented.

The SSH implementation supports both public-key (ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521) and password-based access mechanisms for user authentication through SSHv2. Administrators must associate a public key with each user account that will be authenticated with public-keys. When an SSH session is established the private key used by the remote user must be appropriate for the login ID which is provided during the SSH login. Only if the remote user possesses the correct private key for the login ID will the SSH login be successful.

The TSF supports SSHv2 with AES-CTR-128 or AES-CTR-256 for encryption; HMAC-SHA-256 or HMAC- SHA-512 for integrity; and ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521 for key exchange.

The TSF supports the same public key algorithms (ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521) for host authentication through SSHv2 as for user authentication.

If the TSF's implementation of SSH receives an "SSH packet" larger than 262127 bytes from the TCP layer of the network stack, the TSF silently drops the packet. The TSF uses the packet length field in the SSH header to determine the packet length.

The TSF initiates a rekey if 1GB of data is encrypted with a symmetric encryption key or shortly before 1 hour has elapsed, whichever occurs first.

TLS Client Protocol with Mutual Authentication

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

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The TSF sends the Supported Elliptic Curves extension with secp384r1 in its Client Hello message. The TSF allows the configuration of the supported curves.

The TSF verifies the remote TLS server's certificate is valid described in Section 6.3.

If the Syslog server is specified in the TOE configuration using a DNS name, the TSF automatically generates DNS-ID and CN-ID reference identifiers containing the specified DNS name that will be compared to values within the certificate received from the Syslog server. When the certificate presented by the Syslog server contains the SAN extension the TSF compares the DNS-ID from the TOE configuration to the DNS Name SAN fields. Otherwise, the TSF compares the CN-ID to the CN field(s). In both cases, the TSF performs the comparison as specified in Section 6.4 of RFC 6125, including support for a wildcard in the left-most label of the domain name.

The TOE allows the administrator to determine the behavior of the TSF in the case that the certificate validation or reference identifier matching fails. The flag "allow-crl-failure" may be used to allow or deny connections when the certificate revocation status cannot be verified. In all other cases of certificate validity failure, the TOE does not establish the connection.

If the Syslog server is specified in the TOE configuration using an IP address, the TSF automatically generates IP-ID and CN-ID reference identifiers containing the specified IP address that will be compared to values within the certificate received from the Syslog server. When the certificate presented by the Syslog server contains the SAN extension, the TSF verifies the IP-ID exactly matches an IP address SAN field. If there is not a SAN extension, the TSF will not establish a connection if the CN contains an IP address.

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 TOE supports the following secret keys, private keys and CSPs:

Key	Description	Generation	Non-VolatileStorage	Non-VolatileDestruction	Volatile Destruction
IKE-Pri	VPN Gateway X.509 RSA or ECDSA Private Key	FCS_CKM.1 & FCS_CKM.1/IKE – OpenSSL	Stored in underlying file system	Delete of key by logically addressing the storage location of the key via underlying filesystem APIs and performing a 2-pass overwrite consisting first of a pseudo- random pattern using the TSF's RBG, followed by zeroes	Overwrite with zeros upon shutdown of TOE/VPN
IKE-DH	IKE DH or ECDH Private Key	FCS_CKM.1 & FCS_CKM.2 – QuickSec	N/A	N/A	Overwrite with zeros upon completion of session establishment or rekey

IKE-Ses	IKE AES and HMAC Session Keys	IKE KDF	N/A	N/A	Overwrite with zeros upon session termination or rekey
ESP-Ses	IPsec ESP ES and IMAC Session Keys	IKE KDF	N/A	N/A	Overwrite with zeros upon session termination or rekey
SSH-DH	SSH ECDH Private Key	FCS_CKM.1 & FCS_CKM.2 – OpenSSL	N/A	N/A	Overwrite with zeros upon session establishment or rekey
SSH-Pri	SSH ECDSA Private Host Key	FCS_CKM.1 & FCS_CKM.2 – OpenSSL	Stored in underlying filesystem	Delete of key by logically addressing the storage location of the key via underlying filesystem APIs and performing a two-pass overwriteconsisting first ofa pseudo-random pattern using the TSF's RBG, followed by zeroes	Overwrite with zeros upon shutdown of the TOE/SSH
SSH-Ses	SSH AES and HMAC Session Key	SSH KDF	N/A	N/A	Overwrite with zeros upon session termination or rekey
Sys-Pri	Syslog X.509 TLS RSA or ECDSA Client Private Key	FCS_CKM.1 – OpenSSL	Stored in underlying filesystem	Delete of key bylogically addressing the storage locationof the key via underlying filesystem APIs and performing a two-pass overwriteconsisting first ofa pseudo- random pattern using the TSF's RBG, followedby zeroes	Overwrite with zeros upon session establishment
Sys-DH	TLS ECDH Private Key	FCS_CKM.1 & FCS_CKM.2 – OpenSSL	N/A	N/A	Overwrite with zeros upon session establishment
Sys-Ses	TLS AES Session Keys	TLS KDF	N/A	N/A	Overwrite with zeros upon session termination
DRBG	Internal Stateof DRBG	FCS_RBT_EXT.4 – QuickSec	N/A	N/A	Overwrite with zeros upon shutdown
DRBG	Internal Stateof DRBG	FCS_RBT_EXT.4 – OpenSSL	N/A	N/A	Overwrite with zeros upon shutdown

Table 7 Keys and CSP Information

Summary

The TOE supports key generation and key establishment schemes as shown in Table 8

Scheme	Protocol	Service	Non-Volatile Storage	SFR
RSA key establishment, ECC key establishment, FFC Safe Primes key establishment	IPsec	VPN Gateway (Initiator or Responder)	RAM	FCS_IPSEC_EXT.1
ECC key establishment	SSH	Remote Administration (Server)	RAM	FCS_SSHS_EXT.1
ECDH key establishment	TLS	Syslog (client)	RAM	FCS_TLSC_EXT.1

Table 8 Service, Protocol and Key Establishment Scheme Mapping

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP22e:FCS_CKM.1: The TOE implements asymmetric key generation supporting RSA key establishment (key size 2048/3072), ECC key establishment (curves P-256, P-384, and P-521), and FFC Safe Primes key establishment for IPsec communication as described in the section above. The TOE implements ECC key generation & Key establishment (with curves P-256, P-384, and P-521) as part of the TOE SSH Server. The TOE acts as a client for TLS, providing key generation as support for ECC Key establishment (with curves P-256, P-384, and P-521).
- VPNGW12:FCS_CKM.1/IKE: The TOE generates ECDSA P-256, P-384 and P-521 Elliptic Curve keys as specified in FIPS Pub 186-4 "Digital Signature Standard (DSS)" Appendix B.4 and implements all "shall" and "should" statements and does not implement any "shall not" or "should not" statements.
- NDcPP22e:FCS_CKM.2: See NDcPP22e:FCS_CKM.1.
- NDcPP22e:FCS_CKM.4: The keys and CSP shown in Table 9 are either zeroized or overwritten with a new value when they are no longer needed by the TOE. Zeroization occurs as follows: 1) when deleted from FLASH, the previous value is overwritten once with zeroes; 2) when added or changed in FLASH, any old value is overwritten completely with the new value; and, 3) the zeroization of values in RAM is achieved by overwriting once with zeroes. FLASH and RAM are accessible only through the restricted CLI command set, which can be used only after successful login to the TOE. Since this restricted CLI command set does not offer any commands to view raw FLASH or RAM, the CSP identified in Section 6.2 cannot be viewed even by administrative users.
- NDcPP22e/VPNGW12:FCS_COP.1/DataEncryption: The TOE performs encryption and decryption using AES in CBC, CTR, and GCM mode with key sizes of either 128 or 256.
- NDcPP22e:FCS_COP.1/Hash: The TOE supports cryptographic hashing services using SHA-256, SHA-384, and SHA-512 with message digest sizes 256, 384, and 512. SHS hashing supports keyed-hashing and is used within several services including, VPN Communication, TLS-protected syslog, and SSH-protected remote administration. SHA-256 is used in conjunction with RSA signatures for verification of software image integrity.
- NDcPP22e:FCS_COP.1/KeyedHash: The TOE supports keyed-hash message authentication using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 using SHA-256/384/512 with 256/384/512 bit keys and blocks to produce a 256/384/512 output MAC.
- NDcPP22e:FCS_COP.1/SigGen: The TOE supports the use of RSA with 2048 and 3072 bit key sizes, and ECDSA with a key size of 256 bits or greater for cryptographic signatures (specifically NIST curves P-256, P-384, or P-521).
- NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1: The TOE supports IPsec when communicating with VPN clients and peers.
- NDcPP22e:FCS_RBG_EXT.1: The TOE uses one hardware-based entropy source to seed a software-based DRBG that complies with Special Publication 800-90 using CTR_DRBG. AES-256 is used in conjunction with a minimum of 256 bits of entropy for the seed.
- NDcPP22e:FCS_SSHS_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- NDcPP22e:FCS_TLSC_EXT.1/2: The TOE supports TLS when exporting audit logs to an external server. Certificate pining is not supported. The TOE supports FQDN reference identifiers from RFC 6125 and IPv4 addresses in the SAN. The TOE has support for wildcards in the left-most label of the domain name. The TOE 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.

6.3 Identification and authentication

When a user connects to the console interface, the TSF prompts the user for a username and password. The TSF does not echo any characters back to the local console while the user is entering their password. If the username/password match an authorized administrator's credentials, the user is granted access to the command line interface.

When a user connects to the SSH interface, the TSF checks to see if the user proposed public key authentication. If the client proposed public key authentication, the TSF attempts to authenticate the user using the username and public key authentication. If the public key authentication fails or the client did not propose public key authentication, the TSF attempts to authenticate the client using a username/password. If either SSH_RSA authentication or username/password match an authorized administrator's credentials, the user is granted access to the command line interface.

Prior to authentication at the local or remote console, the TSF allows only the following actions by non-authenticated entities:

- Display the warning banner in accordance with FTA_TAB.1;
- Respond to ICMP Echo Request with an Echo Reply;
- Respond with ICMP Destination Unreachable messages.

Note that a non-authenticated entity which sends an IKE initiation message or an SSH hello message are considered to have initiated the authentication and identification process

The TSF maintains a separate failed authentication counter and lock flag for each remote (SSH) administrative user account. Accounts are never locked out from login at the local console. When a user attempts to establish an SSH session, the TSF checks if the lock flag has been set once the user has provided their username. If the account has been locked, the TSF does not process the authentication data and terminates SSH connection. Otherwise, the TSF processes the authentication attempt. A failed public key authentication attempt immediately followed by a failed username/password authentication attempt in the same SSH session is counted as a single failure, because most SSH clients automatically attempt public key authentication. A failed public key authentication attempt that is not followed by a username/password attempt is still counted as a failed authentication attempt. Each failed username/password authentication attempt is individually counted, with the exception of the case noted above. For each unsuccessful authentication attempt, the TSF increments the counter, compares the counter to the configured limit, and sets the lock flag if the counter has reached the configured limit. For each successful authentication attempt, the TSF resets the failed authentication counter to zero. Accounts can be unlocked only by a user over the local console connection.

The TOE supports a 'root' account that is only allowed to login through the local console. Remote (SSH) login attempts to the root account are always rejected and cannot lock the 'root' account. Thus, the 'root' account can always login at the console and unlock other locked accounts.

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

1. Current date between the "Valid from" and "Valid to" dates
2. If the certificate specifies a OCSP server or CRL distribution point, the certificate is not revoked:
    a. has status "revoked" from the OCSP server (RFC 6960) (Both IPsec and TLS)
    b. is not included in the CLR obtained from the distribution point (IPsec)
3. If the certificate is used for specific purposes, additional checks are performed on extendedKeyUsage and KeyUsage:
    a. OCSP Signing is verified for any certificate used to sign an OCSP response
    b. CRLsign Key Usage bit is verified for any certificate used to sign a CRL
    c. TLS Server Authentication is verified for the certificate used to authenticate the Syslog server
    d. IPsec Tunnel is verified for the certificates used to authenticate VPN peers
4. The certificate path is valid:
    a. The certificate is signed by a certificate that:
        a. Passes all of the certificate validation rules
        b. Has the 'certificate signing' key-usage extension
        c. Has basic constraints CA=True or the certificate is signed by a trusted Root CA

    or

b. The certificate chain of trust (itself, any intermediary CA certificates) chain to a trusted root CA. Root CA and Intermediary CA certificates must be uploaded to the certificate store in the TOE.

The TSF verifies the validity of a certificate when an administrator loads a certificate into the TSF, when the TSF loads its certificates into memory, when IKEv2 receives a client certificate, and when Syslog/TLS receives a server certificate. Administrators may import a trusted root certificate, only if the certificate is self-signed and the certificates has the basicConstraints flag value of TRUE.

The TSF allows the administrator to configure whether the certificate should be accepted or rejected during IPsec-protected VPN communication, if both the CDP and OCSP server do not respond in Step 2 (above). For TLS-protected syslog connections, the TOE will always reject a certificate if the OCSP responder is not available.

The Identification and authentication function satisfies the following security functional requirements:

- NDcPP22e:FIA_AFL.1: Remote administrator accounts can be locked until they are manually unlocked if the failed login threshold is surpassed.
- NDcPP22e:FIA_PMG_EXT.1: The TSF enforces an administrator configurable password length. The minimum password length may be set to 8 to 64 characters. The TSF supports passwords containing upper and lower case letters, digits and the following special characters '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '"', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?' , '[', '_']', '_', '`', '', '|', '~', '<space>'.
- NDcPP22e:FIA_UAU.7: The TSF does not echo any characters back to the local console while the user is entering their password.
- NDcPP22e:FIA_UAU_EXT.2: The TOE uses local password-based authentication.
- NDcPP22e:FIA_UIA_EXT.1: The TOE does not offer any services or access to its functions, except for the specific ICMP messages and displaying a message of the day banner, without requiring a user to be identified and authenticated.
- NDcPP22e/VPNGW12:FIA_X509_EXT.1/Rev: The TOE performs certificate validation for certificate import and TLS authentication using OCSP as described above. The TOE performs certificate validation for IPsec using OCSP and CRL as specified in RFC 5280 and with validation rules as described above.
- NDcPP22e/VPNGW12:FIA_X509_EXT.2: When validating presented certificates (peer certificates in IPsec, server certificates for TLS), if the revocation status of a certificate cannot be verified because the revocation server is unreachable, the TOE will either accept or reject the certificate based upon the TOE configuration. An administrator may configure the TOE behavior for IPsec-protected, VPN communication and for TLS-protected syslog connections.
- NDcPP22e:FIA_X509_EXT.3: The TSF allows the administrator to generate CSRs which contain :
  - Public Key
  - Common Name
  - Country
  - Email
  - Locality
  - Organization
  - Organization Unit
  - Serial Number (the signing CA may override this serial number; this is merely a requestedserial, and is not the serial number of the TOE)
  - State
  - Subject Alternative Name

  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.

6.4 Security management

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

The TSF enforces these restrictions by restricting the administrators to a restricted command environment. When the TSF grants access to an administrative user using SSH protocol or the console, the user has read only access to non-sensitive data. Authorized administrators must run a separate "enable" command, enter an additional password, and be assigned the authorized administrator (exec) privilege to gain access to privileged mode. The TSF generates an additional audit record when a user attempts (successes and failures) to access privileged mode. This audit record includes the username, the time and date and location (remote IP address or console). Any changes to the system configuration or stored data can be performed only in privileged mode.

The TSF restricts the following functions to authorized administrators who have been assigned the security administrator role:

- Configure the access banner
- Configure the remote administrator inactivity timeout
- Configure the local administrator inactivity timeout
- Initiate an update to the software
- Manage the failed authentication lockout threshold
- Start and Stop Services
- Configure audit trail archiving
- Configure Syslog server connectivity
- Manage Cryptographic keys
  - Generate CSR (and RSA or ECDSA key pair)
  - Load a private key (associated with an X.509 certificate)
  - Generate SSH Host Key (ECDSA key pair)
- Manage IPsec security parameters
  - Configure IKEv2 SA lifetimes
  - Configure IKEv2 algorithms
  - Mange IKEv2 Session Establishment restrictions
  - Configure IPsec ESP algorithms
  - Configure IP address assignment to VPN clients
- Unlock account (local console only)
- Manually set the time
- Ability to configure the reference identifier for the peer
  - Configure IP address assignment to VPN clients
  - Configure the Syslog server's reference identifier
- Manage trusted CAs
  - Import X.509 Certificates into the TOE's trust store and designate X509.v3 certificates as trust anchors
  - Delete trust anchor certificates from the TOE's trust store
- Import an X.509 Certificate for use by the TOE
- Load and assign SSH public key for user authentication
- Manage administrator accounts
- Manage minimum password length
- Configure Packet filtering rules
  - Defining Packet filtering rules
  - Ordering Packet filtering rules
  - Assigning Packet filtering rules to interfaces

The Security management function satisfies the following security functional requirements:

- NDcPP22e:FMT_MOF.1/ManualUpdate: Only the authorized administrator can update the TOE.
- NDcPP22e:FMT_MTD.1/CoreData: Only the authorized administrator can configure TSF-related functions.
- NDcPP22e:FMT_MTD.1/CryptoKeys: Only the authorized administrator can configure cryptographic keys. The keys an authorized administrator can manage consist of importing trusted Root CA certs, generating SSH host keys, importing SSH public keys, and loading X.509 certificates. All of these keys can be also be deleted.
- VPNGW12:FMT_MTD.1/CryptoKeys: The TOE permits the management of VPN related cryptographic keys.
- NDcPP22e:FMT_SMF.1: The TOE allows the administrator to perform the administrative functions identified above.
- VPNGW12:FMT_SMF.1/VPN: The TOE allows the administrator to perform packet filtering related management including definition of rules, ordering of rules, and associating rules with interfaces.
- NDcPP22e:FMT_SMR.2: The TOE maintains a single role for all administrative user.

6.5 Packet Filtering

The TSF has three logical network interfaces:

- VPN Ingress (public untrusted network)
- VPN Egress (internal trusted network)
- Management

While the TSF is powering up, the TSF does not enable any network interfaces 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 processes any network data. Once the network interfaces have been enabled, the TSF verifies that all packets which are destined to the TOE are processed using the packet filtering rulesets. Packet filtering rulesets are integral to the correct functioning of the network packet routing functionality; any failure of the packet filtering component will also cause networking to fail. At any time that networking functionality is enabled, the packet filtering rules will be applied.

The TSF implements three different rule chains that can be applied to network traffic on the VPN Ingress, VPN Egress, or Management. Each chain is applied to a different traffic type; traffic addressed to the TOE (INPUT), traffic sent by the TOE (OUTPUT), and traffic passing through the TOE (FORWARD). The rules are applied in the order they appear. Each rule can be ACCEPT, DROP, or LOG. Traffic can be filtered by interface (based on the name of the interface), IP protocol (TCP, UDP), port range, and IP address range. FORWARD rules are applied to all traffic not addressed to the TOE, including decrypted VPN traffic.

The TSF implements support for IPv4, IPv6, 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 TSF implements SPD BYPASS rules using FORWARD rules from one physical interface to another physical interface.

The TSF implements SPD PROTECT rules using FORWARD rules from a physical interface to the VPN. Since the FORWARD rules specify the VPN, a failure of the VPN results in the packets becoming undeliverable. If the VPN is functional but an SA has not been established, the TSF attempts to establish the SA by sending an IKE_SA_INIT message to the peer.

The TSF implements SPD DISCARD rules using a DROP rule on any of the rule chains. The TSF implements one hard-coded iptables rule that cannot be modified:

- DROP (and optionally LOG) any packets that are not matched by previous administrator- configured rules.

Packets entering the TSF's network stack are filtered by a TOE filtering mechanism known as iptables. The iptables mechanism examines the following fields within the header of each packet: Source Address (IPv4 or IPv6), Destination address (IPv4 or IPv6), Protocol (IPv4 or IPv6), Source Port (TCP or UDP), and destination port (TCP or UDP). The IPsec engine then performs its own filtering and processing as necessary to encrypt and decrypt packets. Finally, the resulting packets are processed via iptables once more.

The TSF initially sets iptables to block traffic on the ingress and egress ports. When the VPN service starts, it initializes and performs various self-tests. Once complete, the TOE loads iptables with the active VPN configuration to allow VPN traffic to commence.

The Packet Filtering function satisfies the following security functional requirements:

- VPNGW12:FPF_RUL_EXT.1: The TOE supports all of the required protocols, ipv4 (RFC 791), ipv6 (RFC 2460), tcp (RFC 793), and udp (RFC 768) as well as source and destination address as determined by testing with known good implementations. The TOE does not impose limitations upon the protocols supported by IPv4 and IPv6. The SPD entries implement permit and deny possibilities. Each SPD entry can be configured to log status of packets pertaining to the entry. The TOE supports Ethernet interfaces using the same packet filtering policy mechanism. Administrators can configure rules and apply them to individual interfaces, but all use the same underlying mechanism.

6.6 Protection of the TSF

The TOE has several self-protection mechanisms. Administrator passwords are stored salted and hashed 5000 times with SHA-512. The CLI does not provide the user with any commands that allow for the reading of the hashed passwords. Additionally, the CLI does not provide the user with any commands that allow for the reading of the secret and private keys.

When the TSF starts-up, it runs the following self-tests:

- FIPS Self-Test for the Kernel, OpenSSL, and libgcrypt Cryptographic Modules
  - HMAC Integrity Check

  - Known Answer Tests of cryptographic algorithms
- Entropy Health Check
  - Verification that RDSEED does not report a failure
- Digital Signature verification of all software components

Software POST consists of FIPS self-testing for cryptographic modules, an HMAC integrity check, and the KAT of all cryptographic algorithms.

The RDSEED entropy source performs its own internal status checks continuously. The TSF checks whether the RDSEED instruction reports it is healthy on each call. If the RDSEED indicates a health check failure, the TSF generates an audit record and fails the test. When the test fails, the TSF discards the value read from RDSEED.

The Kernel, OpenSSL, and libgcrypt cryptographic libraries identified in Section 6.2, each perform integrity test using an HMAC and a Known Answer Test (KAT) on each algorithm implemented within it. Each KAT consists of calling the algorithm with known inputs and verifying that the output matches the expected (pre- computed) output.

The libgcrypt cryptographic library is used for hashing used in HMAC operations that support TOE trusted update and product integrity verification. Table 6 in [ST] shows the algorithms supported by libgcrypt. Also, the libgcrypt cryptographic library is used to verify RSA digital signatures during TOE trusted updates and product integrity verification operations.

The TSF verifies the integrity of all TSF components by verifying a digital signature (RSA 2048) of the when each software component is loaded or reloaded into memory for execution. The TSF verifies the integrity of configuration files by comparing a hash of each configuration file to a hash generated when the configuration file was last updated.

If any self-testing generates a failure, the TOE immediately fails-secure by powering off and shutting 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.

The “show system version” command allows the user to query the current overall software version of the TOE.

The TSF utilizes the Red Hat RPM package management system for software updates. The TSF is configured to trust updates that are digitally signed by Red Hat’s private key and Apriva’s private key. The TSF disallows the user from performing an update using a solely Red Hat signed RPM. The update package must be signed by Apriva, but sub-packages can be signed by Red Hat only. This ensures that users do not load arbitrary Red Hat RPMs on the TSF. Both the Red Hat and Apriva private keys are RSA 2048-bit keys. Red Hat-supplied RPM files are digitally signed with Red Hat’s private key. Apriva-created packages are only signed with an Apriva private key. The TSF automatically verifies the signature of any package that is updated. If the signature verification fails, the TSF logs the failure, aborts the update, and deletes the invalid package. If the signature verification succeeds, the TSF installs the update. All updates are “atomic” once the update process has started, so the TOE automatically restarts and activates the update if it was successful, or the TOE rolls back to the previous version if the update failed for any reason.

The TSF stored public keys used to verify software update in plaintext on the file system. The TSF’s CLI prevents the users from modifying these public keys by preventing direct file system access. The only method of modifying a public key is to use the trusted update function to update the package that contains the key.

The Protection of the TSF function satisfies the following security functional requirements:

- NDcPP22e:FPT_APW_EXT.1: The TOE does not offer any functions that will disclose to any user a plain text password. Furthermore, locally defined passwords are not stored in plaintext form

- VPNGW12:FPT_FLS.1/SelfTest: If any self-testing generates a failure, the TOE immediately fails-secure by shutting down.

- NDcPP22e:FPT_SKP_EXT.1: The TOE does not include any user commands or system functions that will disclose any stored cryptographic keys. Keys are stored as identified in **Table 7** when they are created and are stored in an encrypted partition. The key to unlock this partition is stored in a Root only accessible partition and is automatically read by the system during the boot up process.

- NDcPP22e:FPT_STM_EXT.1: The TOE includes its own hardware clock. The TOE allows the administrator to set time manually. The date and time are used as the time stamp that is applied to TOE generated audit records, used to track inactivity of administrative sessions, and perform certificate expiration checks.

- NDcPP22e/VPNGW12:FPT_TST_EXT.1: The TOE performs a suite of self-tests to verify its integrity and proper cryptographic functionality.

- VPNGW12:FPT_TST_EXT.3: The TOE performs a suite of self-tests to verify its integrity.

- NDcPP22e/VPNGW12:FPT_TUD_EXT.1: The TOE provides function to query the version and upgrade the software embedded in the TOE appliance. When installing updated software, the TOE uses digital signatures to authenticate the update to ensure it is the update intended and originated by the vendor.

6.7 TOE access

The administrator can access the TSF via the local console (serial) or remotely via SSH. The TSF displays a configurable advisory and consent message when administrator accesses the CLI through either interface. The administrator can terminate their own CLI session (both local console and SSH) by logging out. The TSF terminates local console sessions and SSH sessions after a configurable period of inactivity.

The TOE access function satisfies the following security functional requirements:

- NDcPP22e:FTA_SSL.3: The TOE terminates remote sessions that have been inactive for an administrator-configured period of time. After termination, administrative authentication is required to access any of the administrative functionality of the TOE.
- NDcPP22e:FTA_SSL.4: The TOE provides the function to logout (or terminate) both local and remote user sessions as directed by the user.
- NDcPP22e:FTA_SSL_EXT.1: The TOE terminates local sessions that have been inactive for an administrator-configured period of time. After termination, administrative authentication is required to access any of the administrative functionality of the TOE.
- NDcPP22e:FTA_TAB.1: The TOE can be configured to display administrator-defined advisory banners when administrators establish interactive sessions with the TOE, allowing administrators to terminate their session prior to performing any functions. The banner is displayed after the login ID has been presented, but before the password prompt is shown.
- VPNGW12:FTA_TSE.1: When acting as a VPN Headend and authenticating VPN users, the TSF can be configured toprevent access based on remote IP address, time of day, and/or day of week.
- VPNGW12:FTA_VCM_EXT.1: The TSF supports the capability of assigning a private IP address to VPN clients upon successful establishment of a session.

6.8 Trusted path/channels

The TOE provides a trusted path for its remote administrative users accessing the TOE using SSH protected Command Line Interface. Local administrators may also use the command line interface through a locally attached console.

Remote connections to third-party syslog servers are supported for exporting audit records to an external audit server. Communication with those external servers is protected using TLS. The TOE acts as a TLS client in communication with a syslog server. Connection to VPN clients and peers is also supported using IPsec.

The Trusted path/channels function satisfies the following security functional requirements:

- NDcPP22e:FTP_ITC.1: The TOE uses TLS when exporting audit records to a third party syslog server. The TOE uses IPsec when communicating with IPsec clients and peers. The TOE provides assured identification of non-TSF endpoints (for both TLS and IPsec) by validating X.509 certificates. The TOE implements a trust store containing trust anchors which it uses to verify identities of those non-TSF certificates.
- VPNGW12:FTP_ITC.1/VPN: When making connections with remote VPN clients and peers the TOE can act as the initiator or the responder for IPsec communication between IPsec peers.
- NDcPP22e:FTP_TRP.1/Admin: The TOE uses SSH to provide a trusted path for remote management interfaces to protect the communication from disclosure and modification.