Architecture Technology Corporation Machete Router Security Target

Version 0.6
11/29/2023

Prepared for:

Architecture Technology Corporation

9971 Valley View Road
Eden Prairie, MN 55344

Prepared By:

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 Machete Router provided by Architecture Technology Corporation. The TOE is being evaluated as a Network Device and 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 – Architecture Technology Corporation Machete Router Security Target

ST Version – Version 0.6

ST Date – 11/29/2023

1.2 TOE Reference

TOE Identification – Architecture Technology Corporation Machete Router

TOE Developer – Architecture Technology Corporation (ATCorp)

Evaluation Sponsor – Architecture Technology Corporation (ATCorp)

1.3 TOE Overview

The Target of Evaluation (TOE) is the Architecture Technology Corporation Machete Router. Machete is a ruggedized, compact, secure and high-performance router that also provides VPN gateway functionality. The functions of Machete are implemented in a software suite called ATCorp Routing and Encryption Suite (ARES).

1.4 TOE Description

The TOE consists of the following hardware:

Model Identification	Platform	CPU Architecture	CPU Part Number
MACHETE-FIT2	Fitlet2	Intel Apollo Lake	Atom x7-E3950
MACHETE-OTN4	OnTime 4000 Series	Intel Apollo Lake	Atom x7-E3950
MACHETE-OTN6	OnTime 6000 Series	Intel Apollo Lake	Atom x7-E3950
MACHETE-OTN7	OnTime 7000 Series	Intel Apollo Lake	Atom x7-E3950
MACHETE-DCS2	DCS003289	Intel Apollo Lake	Atom x7-E3950
MACHETE-V1	VMware ESXi v7.0	AMD Ryzen 4000	Ryzen 4600G
MACHETE-AMD-R1	OL-ML100 Series	AMD Ryzen V1000	V1605B
MACHETE-WL1	BKNUC8V5PNB	Intel Whiskey Lake	Core i5-8365U
MACHETE-FIT3	Fitlet3	Intel Elkhart Lake	Atom x6425E

Running:

- ARES v2.0

1.5 Physical Boundaries

Each TOE appliance runs version 2.0 of the ARES 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.

The TSF requires the following equipment/services to be present in the operational environment:

- IPsec Peers
- CRL server
- Syslog server reachable through either an SSH client connection or through a VPN connection
- NTP server reachable through a VPN connection
- Administrative SSH client
- Web Browser for ESXi Console Management

1.6 Logical Boundaries

This section summarizes the security functions provided by the Machete Router:

- 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 capable of auditing all required events and information. Each audit record includes the identity of the user that caused the event (if applicable), date and time of the event, type of event, and the outcome of the event.

The TOE protects storage of audit information from modification or deletion. The TOE can transmit audit records to a remote syslog server using either SSH or IPsec.

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 and SSH.

1.6.3 Identification and authentication

The TOE supports passwords consisting of alphanumeric and special characters. The TSF also allows administrators to set a minimum password length of 6 to 100 characters.

The TOE requires all administrative users to authenticate before allowing the user to perform any actions other than:

- Viewing the warning banner.

After an administrator-specified number of failed attempts, the user account is locked out. The TOE also protects, stores and allows authorized administrators to load X.509.v3 certificates for use to support authentication for IPsec connections.

1.6.4 Security management

The TOE provides a custom CLI that allows users with the Security Administrator role to administer the TOE locally and remotely. This interface allows the Security Administrator to initiate manual updates, manage cryptographic keys, manage the TOE configuration, and configure audit data transmission.

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 prevents the reading of secret keys, private keys, and passwords.

The TOE maintains a local real-time clock to provide accurate timestamps. This clock can be periodically updated by synchronizing with an NTP server and/or manually set by a Security Administrator.

The TOE performs a suite of power-up self-tests that verify the correct operation of the entropy source, RAM, and cryptographic algorithms as well as the integrity of the firmware.

The TOE verifies the authenticity and integrity of all firmware updates using ECDSA signature verification. The TOE shuts down if any of these tests fail.

1.6.7 TOE access

Before establishing an administrative session, the TOE displays an administrator configurable warning banner. The TOE locks inactive local administrative sessions and terminates inactive remote administrative sessions.

The TOE allows the administrator to configure restrictions on the establishment of client IPsec tunnels based on the client IP address, time of day, date, day of week, or day of month. The TOE assigns a private IP address (internal to the trusted network for which the TOE is the headend) to a VPN client upon successful establishment of a session.

1.6.8 Trusted path/channels

The TOE supports either SSH or IPsec to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data. The TOE uses SSH or IPsec to provide the trusted path with remote administrative users as well.

1.7 TOE Documentation

Machete Router Common Criteria Operational Guidance, Version 1.5, December 12, 2023

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 (CFG_NDcPP-VPNGW_V1.2), which includes the following components:
    - Base-PP: collaborative Protection Profile for Network Devices, Version 2.2e (CPP_ND_V2.2E)
    - PP-Module: PP-Module for VPN Gateways, Version 1.2 (MOD_VPNGW_V1.2)

Package	Technical Decision	Applied	Notes
CPP_ND_V2.2E	TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
CPP_ND_V2.2E	TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
CPP_ND_V2.2E	TD0790: NIT Technical Decision: Clarification Required for testing IPv6	No	Requirement not claimed
CPP_ND_V2.2E	TD0738 - NIT Technical Decision for Link to Allowed-With List	Yes	
CPP_ND_V2.2E	TD0670 - NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	Requirement not claimed
CPP_ND_V2.2E	TD0639 - NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
CPP_ND_V2.2E	TD0638 - NIT Technical Decision for Key Pair Generation for Authentication	Yes	
CPP_ND_V2.2E	TD0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH	Yes	
CPP_ND_V2.2E	TD0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters	No	Requirement not claimed
CPP_ND_V2.2E	TD0632 - NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
CPP_ND_V2.2E	TD0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
CPP_ND_V2.2E	TD0592 - NIT Technical Decision for Local Storage of Audit Records	Yes	
CPP_ND_V2.2E	TD0591 - NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
CPP_ND_V2.2E	TD0581 - NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
CPP_ND_V2.2E	TD0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	

Package	Technical Decision	Applied	Notes
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	Requirement 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	Requirement not claimed
CPP_ND_V2.2E	TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	Requirement 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	Requirement not claimed
CPP_ND_V2.2E	TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	Requirement not claimed
CPP_ND_V2.2E	TD0536 - NIT Technical Decision for Update Verification Inconsistency	Yes	
CPP_ND_V2.2E	TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
CPP_ND_V2.2E	TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
MOD_VPNGW_V1.2	TD0771 - Correction to FIA_PSK_EXT.3 EA	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	Yes	

For formatting purposes, the following acronyms will be used throughout this document:

CPP_ND_V2.2E = NDcPP22e

MOD_VPNGW_V1.2 = VPNGW12

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 VPNs and as such are applicable to the Machete Router 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.CONNECTIONS
The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

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:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0633
- VPNGW12:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0657
- NDcPP22e:FCS_NTP_EXT.1: NTP Protocol
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_SSHC_EXT.1: SSH Client Protocol - per TD0636
- NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
- NDcPP22e:FIA_PMG_EXT.1: Password Management
- VPNGW12:FIA_PSK_EXT.1: Pre-Shared Key Composition
- VPNGW12:FIA_PSK_EXT.2: Generated Pre-Shared Keys
- VPNGW12:FIA_PSK_EXT.3: Password-Based Pre-Shared Keys
- 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:FIA_X509_EXT.2: X.509 Certificate Authentication
- VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication
- 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 the Machete TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP22e/VPNGW12:FAU_GEN.1: Audit Data Generation
	NDcPP22e:FAU_GEN.2: User identity association
	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 TD0633
	NDcPP22e:FCS_NTP_EXT.1: NTP Protocol
	NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP22e:FCS_SSHC_EXT.1: SSH Client Protocol - per TD0636
	NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
FIA: Identification and authentication	NDcPP22e:FIA_AFL.1: Authentication Failure Management
	NDcPP22e:FIA_PMG_EXT.1: Password Management
	VPNGW12:FIA_PSK_EXT.1: Pre-Shared Key Composition
	VPNGW12:FIA_PSK_EXT.3: Password-Based Pre-Shared Keys
	NDcPP22e:FIA_UAU.7: Protected Authentication Feedback
	NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP22e/VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication

Requirement Class	Requirement Component
	NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP22e:FMT_MOF.1/Functions: Management of Security Functions Behaviour
	NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP22e:FMT_MOF.1/Services: Management of Security Functions Behaviour
	NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	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
	VPNGW12:FTA_SSL.3/VPN: TSF-Initiated Termination (VPN Headend) - per TD0656
	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).
- [***selection: no other actions***];

d) Indication that TSF self-test was completed
e) Failure of self-test
f) 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:FCS_COP.1/DataEncryption	None	None
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.	Reason for failure.
	Protocol failures.	Reason for failure. Non-TOE endpoint of connection.
	Establishment or Termination of an IPsec SA.	Non-TOE endpoint of connection.
NDcPP22e:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server
NDcPP22e:FCS_RBG_EXT.1	None	None
NDcPP22e:FCS_SSHC_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP22e:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
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
VPNGW12:FIA_PSK_EXT.1	None	None
VPNGW12:FIA_PSK_EXT.2	None	None
VPNGW12:FIA_PSK_EXT.3	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,	Reason for failure of certificate validation

	replacement or removal of trust anchors in the TOE's trust store	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/Functions	None	None
NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP22e:FMT_MOF.1/Services	None	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
FMT_SMF.1.1/VPN	All administrative actions	No additional information.
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:FPT_TST_EXT.1	None	None
VPNGW12:FPT_TST_EXT.1	None	None
NDcPP22e/VPNGW12:FPT_TST_EXT.1	Execution of TSF self-test.	None.
	Detected integrity violations.	The TSF code file that caused the integrity violation.
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	None

	interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	
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	Termination of the trusted channel Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channel establishment attempt
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.

ST Application Note: *The base FAU_GEN.1 requirement is found in the NDcPP22e base PP. This iteration of FAU_GEN.1 extends the FAU_GEN.1 requirement from NDcPP22e with the additional audit events identified in the VPNGW12.*

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 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: [overwrite the oldest log file]***] 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, P521]] (TD0723 applied)***
and
[***- no other key generation algorithms***]
and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.3 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

NDcPP22e:FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [
- 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 [[a dynamic value that does not contain any CSP]]***];
- 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 [[four]-pass] overwrite consisting of [zeroes, ones, [static 0xAA, static 0x55]]***]

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-1***, ***SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 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-1***, ***HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] and cryptographic key sizes [***160***, **256, 385, 512**] and message digest sizes [***160, 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, 3072 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256-bits, 384-bits, 521-bits]]
that meet the following:
[***- For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 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 [***transport mode, 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)***] 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: [

- ***IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence numbers] and [RFC 4868 for hash functions],***
- ***IKEv2 as defined in RFC 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 [***IKEv1, 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 [

- ***IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [length of time, where the time values can be configured within [0.25 - 24] hours],***
- ***IKEv2 SA lifetimes can be configured by a Security Administrator based on [length of time, where the time values can be configured within [0.25 - 24] hours]***].

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.8

The TSF shall ensure that [

- ***IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [number of bytes, length of time, where the time values can be configured within [0.25 - 8] hours],***
- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [number of bytes, length of time, where the time values can be configured within [0.25 - 8] 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 [**224-bits (Group 14), 256-bits (Group 19), 384-bits (Group 20), 224-bits (Group 24)**] bits.

NDcPP22e/VPNGW12:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv1, IKEv2***] exchanges of length [***according to the security strength associated with the negotiated DH group, 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)] 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 [***IKEv1 Phase 1, IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv1 Phase 2, IKEv2 CHILD_SA***] connection.

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 [***Pre-shared keys***].

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), [***SAN: IP address, SAN: Fully Qualified Domain Name (FQDN)***].

VPNGW12 Application Note: *FCS_IPSEC_EXT.1.4: This SFR element has been modified from its definition in the NDcPP by mandating either 128 or 256 bit key sizes for AES-CBC or AES-GCM, thereby disallowing for the sole*

selection of 192 bit key sizes. When an AES-CBC algorithm is selected, at least one SHA-based HMAC must also be chosen. If only an AES-GCM algorithm is selected, then a SHA-based HMAC is not required since AES-GCM satisfies both confidentiality and integrity functions.

VPNGW12 Application Note: *FCS_IPSEC_EXT.1.11: This element has been modified from its definition in the NDcPP by mandating DH groups 19 and 20, both of which are selectable in the original definition of the element. Any groups other than 19 and 20 may be selected by the ST author but they are not required for conformance to this PP- Module.*

VPNGW12 Application Note*: FCS_IPSEC_EXT.1.14: This PP-Module requires DN to be supported for certificate reference identifiers at minimum. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.*

5.1.2.10 NTP Protocol (NDcPP22e:FCS_NTP_EXT.1)

NDcPP22e:FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [***NTP v4 (RFC 5905)***].

NDcPP22e:FCS_NTP_EXT.1.2
The TSF shall update its system time using [***[IPsec] to provide trusted communication between itself and an NTP time source.***].

NDcPP22e:FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

NDcPP22e:FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.1.2.11 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

NDcPP22e:FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].

NDcPP22e:FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[one] platform-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

5.1.2.12 SSH Client Protocol - per TD0636 (NDcPP22e:FCS_SSHC_EXT.1)

NDcPP22e:FCS_SSHC_EXT.1.1
The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, [***4256, 4344, 5647, 5656, 6668, 8268, 8308 section 3.1, 8332***].

NDcPP22e:FCS_SSHC_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_SSHC_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than [**262,126 bytes**] bytes in an SSH transport connection are dropped.

NDcPP22e:FCS_SSHC_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com***].

NDcPP22e:FCS_SSHC_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa,***

rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521,] as its public key algorithm(s) and rejects all other public key algorithms.

NDcPP22e:FCS_SSHC_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, implicit***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

NDcPP22e:FCS_SSHC_EXT.1.7

The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellmangroup18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

NDcPP22e:FCS_SSHC_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.

NDcPP22e:FCS_SSHC_EXT.1.9

The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [***a list of trusted certification authorities***] as described in RFC 4251 section 4.1.

5.1.2.13 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, [***4256, 4344, 5647, 5656, 6668, 8268, 8308 section 3.1, 8332***].

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 [**262,126 bytes**] 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-cbc, aes256-cbc, aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com***].

NDcPP22e:FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, rsa-sha2-256, rsa-sha2-512, 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-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, implicit***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

NDcPP22e:FCS_SSHS_EXT.1.7

The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellmangroup18-sha512, 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.

5.1.3 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 [**1-20**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

NDcPP22e:FIA_AFL.1.2

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

5.1.3.2 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 [**6**] and [**100**] characters.

5.1.3.3 Pre-Shared Key Composition (VPNGW12:FIA_PSK_EXT.1)

VPNGW12:FIA_PSK_EXT.1.1

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

VPNGW12:FIA_PSK_EXT.1.2

The TSF shall be able to accept the following as pre-shared keys: [***generated bit-based, password-based***] keys.

5.1.3.4 Generated Pre-Shared Keys (VPNGW12:FIA_PSK_EXT.2)

VPNGW12:FIA_PSK_EXT.2.1

The TSF shall be able to [***accept externally generated pre-shared keys, generate [128, 256] bit-based pre-shared keys via FCS_RBG_EXT.1.***].

5.1.3.5 Password-Based Pre-Shared Keys (VPNGW12:FIA_PSK_EXT.3)

VPNGW12:FIA_PSK_EXT.3.1

The TSF shall support a PSK of up to [**130**] characters.

VPNGW12:FIA_PSK_EXT.3.2

The TSF shall allow PSKs to be composed of any combination of upper case characters, lower case characters, numbers, and the following special characters: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')', and [***'+', '/', '-', '_', '=', '?', '<space>'***]

VPNGW12:FIA_PSK_EXT.3.3

The TSF shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC- [***SHA-256***] , with [**10,000**] iterations, and output cryptographic key sizes [***256***] that meet the following: NIST SP 800-132.

VPNGW12:FIA_PSK_EXT.3.4

The TSF shall not accept PSKs less than [**8**] and greater than the maximum PSK length defined in VPNGW12:FIA_PSK_EXT.3.1.

VPNGW12:FIA_PSK_EXT.3.5
The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 and with entropy of [**128**] bits.
VPNGW12:FIA_PSK_EXT.3.6
The TSF shall require the PSK to be entered before every initiated connection.
VPNGW12:FIA_PSK_EXT.3.7
The TSF shall [**perform no action to assist the user in choosing a strong password**].

5.1.3.6 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.

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

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

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

NDcPP22e:FIA_UIA_EXT.1.1
The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:
- Display the warning banner in accordance with FTA_TAB.1;
- [***no other actions***].

NDcPP22e:FIA_UIA_EXT.1.2
The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

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

NDcPP22e:FIA_X509_EXT.1.1/Rev
The TSF shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
o Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

NDcPP22e:FIA_X509_EXT.1.2/Rev
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

VPNGW12 Application Note: FIA_X509_EXT.2.1: *This is specified as a selection-based SFR in the Base-PP but is mandatory for any TOE that claims conformance to this PP-Module because a conformant TOE will always have the ability to receive an X.509 certificate from an external entity as part of IPsec communications. Therefore, a mechanism for the TSF to validate an X.509 certificate presented to it is required*

5.1.3.10 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 [***no other protocols***], 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***].

VPNGW12 Application Note: FIA_X509_EXT.2.1: *The Base-PP allows the ST author to specify the TSF's use of X.509 certificates. Because this PP-Module mandates IPsec functionality, the SFR has been refined to force the inclusion of it. Other functions specified by the Base-PP may be chosen without restriction.*

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

NDcPP22e:FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, 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.

VPNGW12 Application Note: *The Base-PP defines this SFR as selection-based with its inclusion being dependent on the communications protocols that the TSF supports. Since a TOE that conforms to this PP-Module must support IPsec, this SFR is mandatory. Aside from mandating its inclusion in the TOE boundary, this PP-Module does not modify the SFR.*

5.1.4 Security management (FMT)

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

NDcPP22e:FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [***determine the behaviour of***] the functions [***transmission of audit data to an external IT entity***] to Security Administrators.

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

NDcPP22e:FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

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

NDcPP22e:FMT_MOF.1.1/Services
The TSF shall restrict the ability to start and stop services to Security Administrators.

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

NDcPP22e:FMT_MTD.1.1/CoreData

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

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

NDcPP22e:FMT_MTD.1.1/CryptoKeys

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

5.1.4.6 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.

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

NDcPP22e:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to start and stop services,***
- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),
- Ability to manage the cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure the lifetime for IPsec SAs,
- Ability to set the time which is used for time-stamps,
- Ability to configure NTP,
- Ability to configure the reference identifier for the peer,
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to import X509v3 certificates to the TOE's trust store].

5.1.4.8 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***].

5.1.4.9 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 2460)
  - o Source address
  - o Destination Address
  - o Next Header (Protocol)
- TCP (RFC 793)
  - o Source Port
  - o Destination Port
- UDP (RFC 768)
  - o Source Port
  - o Destination Port

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, synchronise time with an NTP server***].

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 start-up (on power on)]***] to demonstrate the correct operation of the TSF: noise source health tests, [**integrity verification, RAM tests, cryptographic algorithm tests**].

VPNGW12 Application Note: *This SFR is modified from its definition in the NDcPP by requiring noise source health tests to be performed regardless of what other testing is claimed. It is expected that the behavior of this testing will be described in the entropy documentation.*

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.

VPNGW12 Application Note: *The NDcPP provides an option for how firmware/software updates can be verified but this PP-Module requires the digital signature method to be selected at minimum*

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 TSF-Initiated Termination (VPN Headend) - per TD0656 (VPNGW12:FTA_SSL.3/VPN)

VPNGW12:FTA_SSL.3.1/VPN

The TSF shall terminate a remote VPN client session after an Administrator configurable time interval of session inactivity.

5.1.7.3 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.

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

NDcPP22e:FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session***] after a Security Administrator-specified time period of inactivity.

5.1.7.5 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.

5.1.7.6 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***].

5.1.7.7 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, SSH***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***IVPN communications, NTP server***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

NDcPP22e:FTP_ITC.1.2
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.
NDcPP22e:FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [**audit server, NTP server**].

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.
VPNGW12:FTP_ITC.1.2/VPN
The TSF shall permit the authorized IT entities to initiate communication via the trusted channel.
VPNGW12:FTP_ITC.1.3/VPN
The TSF shall initiate communication via the trusted channel for [**remote VPN gateways or peers]**

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 [***IPsec, 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 TOE generates audit logs for the events identified in **Table 2**. These audit records include the date/time, description of the event, operator identity, and success/failure. For cryptographic key management events, the TSF includes the identity of the cryptographic key being acted upon (key file name).

For each packet filtering rule configured as described in Section 6.5, the administrator can instruct the TOE to log any traffic matching the rule. These logs include the source and destination addresses, TOE interfaces, Transport layer protocol (if applicable), and source/destination ports (if applicable).

The TOE allocates 2GB of space to local audit storage. The TOE maintains 4 separate log types and up to 10 files for each log type, one active file and 9 archive files. The current file for each log type is allowed to reach 50MB in size before it is rotated, when the TOE attempts to write an audit log message to an audit file which would increase the size beyond 50MB then that audit log type is rotated. Audit log rotation will delete the oldest archive file, if archiving the current file will create more than 9 archive files. Each archive file is renamed with a suffix .1.gz through .9.gz, indicating the age of the archive file, 1 being the newest and 9 being the oldest. Then the current audit log is compressed and renamed with the suffix .1.gz. Once rotation is completed, a new empty current file is created and the queued audit message is written to the file. This approach ensures that no audit log messages are discarded during the log rotation process and that the allocated storage capacity of 2GB is never exceeded.

The CLI does not implement any commands that allow the modification or deletion of audit records (i.e. The TOE does not allow authorized or unauthorized modification or deletion of audit records).

The TOE transmits audit data to an external audit server using the syslog protocol tunneled within IPsec or SSH. The TOE transmits audit records to the syslog server in real-time. The TOE does not have the ability to cache or retransmit audit records if the syslog server is unavailable.

The Security audit function satisfies the following security functional requirements:

- NDcPP22e/VPNGW12:FAU_GEN.1: The TOE generates audit events for the not specified level of audit. Each audit record includes the date and time of the event, type of event, subject identity.
- 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_EXT.1: The TOE can be configured to export audit records to an external SYSLOG server. This communication is protected with the use of IPsec or SSH.

6.2 Cryptographic support

The TOE supports a range of cryptographic services using the OpenSSL version 3.0.12 cryptographic library. The following functions have been CAVP certified:

Functions	Requirement	Certificate #
Encryption/Decryption		
AES CBC, GCM, CTR (128 and 256 bits)	NDcPP22e/VPNGW12 FCS_COP.1/DataEncryption	A4781
Cryptographic signature services		
• RSA Digital Signature Algorithm (rDSA) (2048, 3072 bits) • Elliptic Curve Digital Signature Algorithm (P-256, P-384, P-521)	NDcPP:FCS_COP.1/SigGen	A4781
Cryptographic hashing		
SHA-1, SHA-256, SHA-384, SHA-512	NDcPP22e:FCS_COP.1/Hash	A4781
Keyed-hash message authentication		
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	NDcPP22e:FCS_COP.1/KeyedHash	A4781
Random bit generation		
CTR_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism	NDcPP22e:FCS_RBC_EXT.1	A4781
Key generation		
• RSA Key Generation (2048, 3072 bits) • ECC Key Generation (P-256, P-384, P-521)	NDcPP22e:FCS_CKM.1 VPNGW12:FCS_CKM.1/IKE	A4781
• FFC Safe primes key generation	NDcPP22e:FCS_CKM.1	Tested with a known good implementation
Key establishment		
• ECC KAS	NDcPP22e:FCS_CKM.2	A4781
• RSA Key Generation (2048, 3072 bits) • FFC Safe primes key generation	NDcPP22e:FCS_CKM.2	Tested with a known good implementation

Table 4 CAVP Certificates

The TOE fulfills all of the FIPS PUB 186-4 requirements for cryptographic key generation without extensions. The TOE conforms to all shall, shall-not, should and should-not statements. The TSF generates asymmetric RSA/ECDSA/ECDH keys for IKE key exchange and IKE authentication according to FIPS 186-4 Appendices B.3.3, B.4 and B.4.2. The TSF supports 2048-bit and 3072-bit RSA keys and ECDSA curves P-256, P-384, and P-521 for IKE authentication. The TSF supports curves P-256, P-384 and P-521 for IKE key exchange. The TOE also generates DH keys for IKE key exchange according to FIPS 186-4 Appendices B.1 and B.1.2. The TOE implementation of Diffie-Hellman group 14 (2048 MODP), Diffie-Hellman group 16 (4096 MODP), and Diffie-Hellman group 18 (8192 MODP) meets RFC 3526, Section 3.

All key generation utilizes the evaluated DRBG.

Security Function	Communication Type	Key Establishment Methods
Administration	SSH (server)	RSA Schemes ECC Schemes DH-14, DH-16, DH-18
Trusted Channels for Syslog	SSH (client)	RSA Schemes ECC Schemes DH-14, DH-16, DH-18
Trusted Channels for Syslog and NTP	IPsec	RSA Schemes ECC Schemes DH-14

Table 5 Key Exchange Methods used by TOE Services

The TOE stores all secret and private cryptographic keys in plaintext. The CLI does not implement any commands that allow the administrator to view these keys. **Table 6** lists the secret and private keys stored by the TOE.

Key	Origin	Storage	Destruction
IKE RSA Private Key 2048 bits, 3072 bits	RSA Key Generation Administrator Loaded	RAM Flash	RAM – VPN service shutdown Flash – Key destruction command
IKE ECDSA Private Key P-256, P-384, or P-521	ECDSA Key Generation Administrator Loaded	RAM Flash	RAM – VPN service shutdown Flash – Key destruction command
IKE ECDH Private Key P-256 or P-384	ECDH Key Generation	RAM	Key exchange completion
IKE DH Private Key 2048 bit modp	DH Key Generation	RAM	Key exchange completion
FW Integrity ECDSA Key P-521	FIPS 186-4 ECDSA Key Generation – performed external to the TOE	RAM Flash	RAM – Manifest signature generation Flash – Key destruction command
DRBG State AES-256	DRBG Initialization	RAM	Shutdown/Reboot
IKE Session Keys AES (128 or 256) and HMAC (256, 384, or 512)	IKEv1 or IKEv2 PRF	RAM	Phase 1/IKE SA Rekey Session termination
ESP Session Keys AES (128 or 256) and HMAC (256, 384, or 512)	IKEv1 or IKEv2 PRF	RAM	Phase 2/Child SA Rekey Session termination
SSH host private key	RSA/ECDSA Key Generation Administrator Loaded	Flash	Overwrite with command
SSH host public key	RSA/ECDSA Key Generation Administrator Loaded	Flash	Overwrite with command
SSH client private key	RSA/ECDSA Key Generation Administrator Loaded	Flash	Overwrite with command
SSH client public key	Client provided	Flash	Public data

SSH session key	Generated on negotiation	RAM	Overwrite at session termination

Table 6 Secret and Private Cryptographic Keys

The TSF destroys keys in RAM by writing a dynamic value to the memory address(es) containing the key being destroyed.

The TSF destroys keys in Flash by logically addressing the storage address and overwriting the key with four different static patterns (i.e. 0x00, 0xFF, 0xAA, and 0x55). The TSF performs a read-verify of the 0x55 pattern after the final write. The TSF logs a key destruction error if the read-verify fails.

The TSF destroys keys when indicated in the Destruction column of **Table 6** using the methods describe above without any operating interaction or configuration.

IPsec Protocol

When a packet is received by the TSF and addressed to the TOE, the TSF first checks the packet against the INPUT filter rules as described in Section 6.5. Packets that match a "permit" rule are allowed to be processed further by the TOE. Next, the TSF determines if the packet is an IKE packet (UDP port 500 or 4500). If the packet is an IKE packet, the TSF checks the packet against the VPN session establishment rules. If the packet does not pass the VPN session establishment rules, the TSF drops the packet. If the packet passes the checks, the TSF processes the IKE packet.

All other packets must match a "permit" INPUT rule for the TOE to process them.

If the packet is an ESP packet (IPv4 protocol 50/IPv6 Next Header 50), the TSF examines the SPI and compares it to the SPIs associated with established Phase 2/Child SAs. If the TSF finds a matching SPI, it attempts decryption and processes the decrypted packet according to Section 6.5. The TSF drops the packet if it does not find a matching SPI or the decryption fails.

All packets not addressed to the TOE are first examined by the FORWARD filter rules, if the packet matches a "permit" rule, then the TOE continues processing the packet. Next, the TSF determines if the packet matches any of the configured IPsec SPD rules. If the packet does not match any SPD rules, the packet is processed as described in Section 6.5. If the packet matches one or more SPD rules, the TSF selects the most specific rule and applies the configured action (i.e. PROTECT, BYPASS, or DISCARD). The specificity of the rule is determined first by the specificity of the IP address/mask, then by the TCP or UDP port, and lastly by the transport layer protocol. Source and destination addresses and ports are given equal weights, so the administrator is instructed to ensure rules cannot have equal specificity. Packets matching a BYPASS SPD rule are forwarded without modification on the appropriate network interface to reach their destination. If the VPN connection for a packet matching a PROTECT SPD rule is not currently established, the TSF sends an IKE init packet to the peer to attempt to establish a connection and the original packet is dropped; otherwise, the TSF forwards the packet through the established VPN connection. Packets that are forwarded through the VPN connection are subject to the OUTPUT filter rules described in Section 6.5.

All network traffic must match a "permit" OUTPUT rule. Then, locally originating traffic is matched against any of the configured IPsec SPD rules in the same manner as described above for forwarded traffic.

The TSF implements IKEv1, IKEv2, and ESP to protect IPsec communications. The TSF will not propose aggressive mode when using IKEv1 and rejects any init packet proposing aggressive mode. The TSF supports both transport mode and tunnel mode for ESP.

The TSF supports the following algorithms for IKEv1 and IKEv2:

- Authentication:
  - X.509 Certificate using RSA >= 2048 bits, ECDSA P-256, P-384, or P-521
  - Pre-Shared Key
- Key exchange (Private key "x" is generated as specified in Section 7.2.1):
  - Group 14
  - Group 19
  - Group 20
  - Group 24
- Encryption:

  - AES-CBC-128
  - AES-CBC-256
  - AES-GCM-128 (IKEv2 only)
  - AES-GCM-256 (IKEv2 only)
- Message authentication/PRF:
  - HMAC-SHA-256
  - HMAC-SHA-384
  - HMAC-SHA-512

The TSF supports the following algorithms for ESP:

- Encryption:
  - AES-CBC-128
  - AES-CBC-256
- Message authentication:
  - HMAC-SHA-256
  - HMAC-SHA-384
  - HMAC-SHA-512

The TSF will not load/enable any IPsec configuration where the ESP encryption algorithm is or can be stronger than the IKE authentication algorithm (e.g. IKE is configured to use AES 128 or 256 while ESP is configured to use AES 256). The TSF will reject any init proposal that specifies an ESP algorithm stronger than the IKE algorithm.

The TOE generates the secret value x used in the IKEv1/IKEv2 Diffie-Hellman key exchange ('x' in gx mod p) using the FIPS validated RBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256 or 384 bits (for DH Groups 14 & 24, 19, and 20, respectively). Nonces are generated using the FIPS validated RBG specified in FCS_RBG_EXT.1. The TSF generates nonce that are 32 bytes long, which is half the length of the longest PRF hash (SHA-512) and greater than the strength of the strongest Diffie-Hellman group (Group 20).

When acting as the initiator, the TSF proposes all configured DH groups and verifies that the responder selects a configured group. When acting as the responder, the TSF parses the list of received proposals and selects the first configured proposal (including DH group) that is supported by the TSF's configuration. A proposal is a set of encryption algorithm, encryption key size, PRF, DH group, and integrity algorithm.

The TSF allows the IKEv1 Phase 1 SA and IKEv2 IKE SA lifetimes to be configured to between 15 minutes and 24 hours.
The TSF allows the IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes to be configured to between 15 minutes and 8 hours and/or 10 million bytes and 4 billion bytes, whichever occurs first.

IKEv1 authentication using X.509 certificates with RSA or ECDSA keys is performed as specified in RFC 2409 Section 5.1, while authentication using pre-shared keys is performed as specified in RFC 2409 Section 5.4.

IKEv2 authentication using X.509 certificates with RSA or ECDSA keys and authentication using pre-shared keys is performed as specified in RFC 5996 Section 2.15.

Pre-shared keys are described in 6.3.

The TSF determines if an X.509 certificate is valid by applying the X.509 Validation rules specified in Section 6.3. If the certificate is valid, the TSF attempts to authenticate the connection using the ID Payload sent by the peer or ID configured by the Administrator for the specific peer. If the ID Type is ID_IPV4_ADDR, ID_RFC822_ADDR, or ID_IPV6_ADDR; the TSF attempts to match the ID against an appropriate SAN field in the certificate. If the ID Type is ID_DER_ASN1_DN, the TSF matches the ID against the DN in the certificate.

By default the TSF uses the DN from its own certificate as its Identification data; however, the Administrator can specify a different ID Type and value.

SSH Client and Server Protocol

The TOE supports SSHv2 with AES (CBC/CTR/GCM modes) 128- or 256-bit ciphers, in conjunction with HMAC-SHA-1, HMAC-SHA-1-96, HMAC-SHA2-256, and HMAC-SHA2-512 integrity algorithms as well as RSA and ECDH using the following key exchange methods:

- diffie-hellman-group14-sha1
- ecdh-sha2-nistp256
- diffie-hellman-group14-sha256
- diffie-hellman-group16-sha512
- diffie-hellmangroup18-sha512
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521

The TOE supports user public key authentication using ssh_rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, or ecdsa-sha2-nistp521. Administrators must associate a public key with each user account that is authenticated with public-keys.

The TOE's SSHv2 supports both public-key and password-based authentication. Either authentication approach can be configured. For the SSH Client, the SSH server can be associated with its public key locally. The maximum packet size accepted by the TOE SSH Server/Client is limited to 262,126 bytes. Whenever the timeout period, or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full the packet will be dropped. The TOE initiates a session rekey after the configured rekey limit is reached. This limit is based on time and data, with a rekey triggered whenever either limit is reached. The time-based rekey limit can be configured by an administrator to values between 1 second and 1 hour. The data-based rekey limit can be configured by an administrator to values between 1 kilobyte and 1 gigabyte of traffic.

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP22e:FCS_CKM.1: The TOE supports asymmetric key generation using RSA key establishment (key size 2048/3072) , ECC key establishment (curves P-256, P-384, and P-521), and FFC Safe Primes key establishment as part of IPsec and SSH as described in the section above.
- VPNGW12:FCS_CKM.1/IKE: See NDcPP22e:FCS_CKM.1
- NDcPP22e:FCS_CKM.2: See NDcPP22e:FCS_CKM.1
- NDcPP22e:FCS_CKM.4: Keys are zeroized when they are no longer needed by the TOE
- NDcPP22e:.VPNGW21: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-1, SHA-256, SHA-384, and SHA-512 with message digest sizes 160, 256, 384, and 512. The TOE supports SHA-256 and 384 hashing in support of HMAC and digital signatures, and SHA-512 for HMAC support, digital signatures, password obfuscation, and binary integrity checking.
- NDcPP22e:FCS_COP.1/KeyedHash: The TOE supports HMAC-SHA-1, HMAC-SHA-1-96, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 (key and output MAC sizes 160, 256, 384, and 512, respectively) for keyed-hash message authentication. The TOE implements HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 in support of IPsec and HMAC-SHA-1, HMAC-SHA-1-96, HMAC-SHA2-256, and HMAC-SHA2-512 in support of SSH.
- 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/VPNGW21:FCS_IPSEC_EXT.1: The TOE supports IPsec when communicating with VPN clients and peers and when communicating with audit and NTP servers.

- NDcPP22e:FCS_NTP_EXT.1: The TOE provides the ability to synchronize its time with a NTP server using NTP v4. The time data is protected by an IPsec connection.
- NDcPP22e:FCS_RBG_EXT.1: The TSF implements a NIST SP 800-90A CTR_DRBG with AES-256 for generating random bits. The TSF instantiates the DRBG using 262,144 of data gathered from the RDRAND instruction which is estimated to provide at least 256-bits of entropy.
- NDcPP22e:FCS_SSHC_EXT.1/NDcPP: FCS_SSHS_EXT.1: SSHv2 is implemented as described above. The TOE is a client in support of exporting auditing records to the syslog server and a server in support of remote management.

6.3 Identification and authentication

The logon process for the local and remote administration is the same. The TOE presents the user with the administrator configured banner and requires the user enter a username and password combination. If the username does not exist, the TOE indicates the authentication attempt failed (without specifically indicating the username does not exist). If the username exists; the TOE reads the password salt associated with the username, concatenates the salt with the provided password, SHA-512 hashes the combined value, and compares the resulting hash to the stored hash. If the hashes match, the TOE authenticates the user. If the hashes do not match, the TOE indicates the authentication attempt failed. The TOE allows administrators to authenticate to the local console by entering a username and password.

The TOE accepts pre-shared keys for IPsec. It accepts bit-based pre-shared keys and accepts text-based PSKs that are transformed into bit-based PSKs. For IPsec, text-based keys are conditioned by HMAC-SHA-256, with 10,000 iterations, and output cryptographic key sizes 256 that meet the following: NIST SP 800-132. The TOE can generate bit-based pre-shared keys of size 128 or 256 bits using the evaluated DRBG.

The TOE maintains a counter of successive failed remote authentication attempts associated with each administrator username. The TSF increments this counter when it receives an invalid password. If the counter exceeds the administrator configured threshold (between 1 and 20), the TOE locks the account from logging in remotely, records the lock time, and will not allow any remote authentication attempts to succeed until the administrator configured time period has elapsed. The account remote lock time period can be configured to be 1 to 3600 seconds. While the account is locked, the TOE will not process any remote authentication attempts for the locked account. Any attempt to log in to the remote account before the configured lock period has expired will reset the lock time to the configured time period. The TOE resets the failed authentication counter when the administrator configured time period has elapsed and when a valid password is entered for an unlocked account. Authentication and administration using the local console remains available for accounts that have been locked out from using the remote authentication mechanisms.

The TOE performs the following checks when validating certificates:

- All Certificates:
  - Proper encoding
  - Valid signature
  - Current time is after the Not Valid Before value
  - Current time is before the Not Valid After value
  - (if present in the certificate) CRL Check
- CA Certificates
  - Basic Constraints: CA=true
  - Extended Key Usage:
  - Certificate Sign
  - CRL Sign (for CA certificates used to verify a CRL)
- TOE and Peer Certificates
  - Basic Constraints: CA=false

The TOE validates Certificates when they are loaded onto the TOE, when they are used to authenticate the TOE to a peer, and when they are used to authenticate a peer to the TOE. If any of the checks fail (with the possible exception of the CRL check), the TSF rejects the certificate. The administrator configures the certificate to authenticate the TOE to a peer while configuring IPsec. The TOE uses the certificate presented by a peer to attempt to authenticate the peer.

The TOE allows the administrator to configure whether a certificate will be considered trusted if the CRL cannot be retrieved.

The TOE does not support the Code Signing, Server Authentication, Client Authentication or OCSP signing extended key usage fields.

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

- NDcPP22e:FIA_AFL.1: When the defined number of unsuccessful authentication attempts has been met, the TSF prevents the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed.
- NDcPP22e:FIA_PMG_EXT.1: The TSF supports passwords composed of upper case, lowercase, numbers, and the symbols in the SFR. The TOE be configured to enforce a minimum password length of 6 to 100 characters.
- VPNGW12:FIA_PSK_EXT.1/2/3: The TOE supports text-based pre-shared keys for IKE PSK authentication. The TOE supports keys composed of any combination of uppercase, lowercase, numbers, and the symbols in the SFR. Text-based pre-shared keys can be anywhere from 8 to 130 characters long. The TSF uses SHA-256 condition text-based pre-shared keys. The TSF allows the administrator to use any of the following methods to create a pre-shared key: generated bit-based (externally and onboard using the DRBG), and password-based.
- NDcPP22e:FIA_UAU.7: The TOE does not echo any characters back to the local console while passwords are being entered
- NDcPP22e:FIA_UAU_EXT.2: The TOE uses local password-based and SSH public key-based authentication.
- NDcPP22e:FIA_UIA_EXT.1: The TSF does not provide any services at the local console prior to authentication.
- NDcPP22e:FIA_X509_EXT.1/Rev: The TOE performs certificate validation as described above.
- NDcPP22e/VPNGW12:FIA_X509_EXT.2: The TOE uses X509v3 certificates to support authentication for IPsec. If the TOE cannot determine the validity of a certificate, the administrator has the option of choosing whether or not to accept the certificate. The administrator accomplishes this by setting the certificate revocation policy in the router configuration.
- VPNGW12:FIA_X509_EXT.3: The TSF allows the administrator to generate certificate signing requests. The CSRs include the public key, common name, organization, and country.

6.4 Security management

The TOE requires administrators to be successfully identified and authenticated prior to allowing the administrator to manage the TOE. The TOE allows the administrators to perform the following management functions over the local and remote administrative interfaces:

- 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 manage the cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure the lifetime for IPsec SAs,

- Ability to set the time which is used for time-stamps;,
- Ability to configure NTP,
- Ability to configure the reference identifier for the peer;,
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to import X509v3 certificates to the TOE's trust store
- Definition of packet filtering rules;
- Association of packet filtering rules to network interfaces;
- Ordering of packet filtering rules by priority

The Security management function satisfies the following security functional requirements:

- NDcPP22e:FMT_MOF.1/Functions: The administrator has the ability to configure the transmission of the audit records to a remote server and secure the transmission.
- NDcPP22e:FMT_MOF.1/ManualUpdate: Only the authorized administrator can update the TOE.
- NDcPP22e:FMT_MOF.1/Services: Only the authorized administrator can start and stop services on the TOE. This is performed via a CLI ‘services’ command. This includes the firewall, NTP, SSH server, and VPN services.
- 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 client 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 riles, and associating rules with interfaces.
- NDcPP22e:FMT_SMR.2: The TOE maintains the security role of Security Administrator who can manage the TOE both remotely and locally.

6.5 Packet Filtering

Prior to enabling the kernel networking stack, the TOE assigns a drop rule to each interface, so no network traffic can be processed. As described below, the TOE does not perform packet switching in hardware, so the kernel networking stack is the only method of processing or forwarding packets. Following initialization, the TOE adjusts the rules assigned to each interface to enable the processing, routing, and filtering of network traffic. The TOE implements packet filtering and forwarding in the kernel, so any failure of the packet filtering rules is a kernel networking stack crash which prevents the TOE from processing any network traffic.

The TOE does not include hardware switching capabilities, so all packets must be processed and routed by the CPU. If the CPU is unable to process packets received on a physical interface or a physical interface is unable to send packets processed by the CPU, the TSF drops the excess packets. The TOE records the number of received packets that are dropped as described in Section 6.1.

The TOE supports the following protocols:

- IPv4 (RFC 791)
- IPv6 (RFC 2460)
- TCP (RFC 793)
- UDP (RFC 768)

The correctness of the TOE's implementation of these protocols is demonstrated through interoperability testing with other devices and network services.

The TSF supports packet filtering based on the following fields within each protocol:

- IPv4 (RFC 791)
  - Source address
  - Destination Address
  - Protocol
- IPv6 (RFC 2460)
  - Source address
  - Destination Address
  - Next Header (Protocol)
- TCP (RFC 793)
  - Source Port
  - Destination Port
- UDP (RFC 768)
  - Source Port
  - Destination Port

Each packet filtering rule is configured as a permit (pass packets), deny (drop packets silently), reject (drop packets and issue and ICMP response), or log rule. The TOE applies the packet filtering rules in the order they are configured by the administrator. If a packet does not match any of the configured rules, the TOE drops and logs the packet without sending an ICMP response.

The TOE allows packet filter rules to be associated with one or more interfaces. The TOE does not define any interface groups; however, the TOE allows wildcards to be used when specifying the interfaces a rule applies to. INPUT and OUTPUT rules can be assigned to physical or virtual interfaces. FORWARD rules allow packets to be routed from one interface to another.

With the exception of IKE and ESP packets, all received packets destined for the TOE (based on IP address) are first subjected to the INPUT rules associated with the receiving interface. All routed packets are subjected to the FORWARD rules to determine if the packet should be passed on. All packets generated locally by the TOE are subjected to the OUTPUT rules associated with the sending interface.

The Packet Filtering function satisfies the following security functional requirements:

- VPNGW12:FPF_RUL_EXT.1: The TOE performs Packet Filtering on network packets processed by the TOE. 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. The SPD entries implement permit and deny possibilities. Each SPD entry can be configured to log status of packets pertaining to the entry.

6.6 Protection of the TSF

The TOE implements several self-protection mechanisms.

During power-up, the TSF runs the following self-tests to verify the correct operation of the TSF:

- Entropy Source Health Tests
  - KAT for the deterministic portion of the entropy source
  - Sliding window test of the first 65536 debiased bits from the entropy source
- RAM Tests on a random 64 MB sample of User space RAM:
  - Random Value – detect bad bits, which are permanently stuck
  - Sequential Increment – detect bad bits, which do not consistently hold a value
  - Walking Ones – detect bad bits, which are dependent on the current values of surrounding bits
  - Stuck Address – determine if the memory locations are addressed properly
- Firmware Integrity Test
  - ECDSA P-521 with SHA-512 signature verification of the firmware manifest and SHA-512 verification of each binary

- Cryptographic Library Tests
  - ECDSA P-224 Pairwise Consistency Test
  - AES GCM 256 Encrypt and Decrypt KATs
  - HMAC-SHA-256 KAT (implicitly tests SHA-256)
  - HMAC-SHA-384 KAT (implicitly tests SHA-384)
  - HMAC-SHA-512 KAT (implicitly tests SHA-512)
  - DRBG

The TSF allows the administrator to query the version of TOE firmware by running the "show version" command.

The TSF supports a manual update process initiated by the administrator. First the administrator must obtain a candidate update from ATCorp on optical media. Then the administrator transfers the update to the TOE, and the TOE verifies the ECDSA P-521 with SHA-512 signature of the update using an ATCorp public key. If the verification fails, the TOE stops the update process and deletes update. If the verification succeeds; the TOE places the update in a staging area, updates the manifest using SHA-512 and the FW Integrity ECDSA Key, and immediately restarts.

The TSF contains a real-time clock. The administrator has the option to either set the time on the clock manually or configure an external NTP server that can be queried periodically to update the clock to ensure that it is accurate and reliable. The TSF utilizes the time for the following security functions:

- Administrative session timeout checking
- Administrative remote authentication lockout timer
- Certificate expiration checking
- Phase 1/IKE SA rekey/expiration interval
- Phase 2/Child SA rekey/expiration interval
- Audit record timestamps
- VPN session inactivity checking
- VPN session establishment checking
- Log rotation

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

- NDcPP22e:FPT_APW_EXT.1: The TOE stores administrator passwords in a non-plaintext form by hashing the password using SHA-512 prior to storage. The CLI does not implement any commands that allow the administrator to view the hashed passwords.
- 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 CLI does not implement any commands that allow the administrator to view pre-shared keys, symmetric keys, and private keys
- NDcPP22e:FPT_STM_EXT.1: The TOE includes its own hardware clock which the administrator can choose to set the time manually or synchronize its time with an external NTP server.
- NDcPP22e/VPNGW12:FPT_TST_EXT.1, VPNGW12:FPT_TST_EXT.3: The TOE offers a suite of self-tests to verify the correct operation of the key generation and static TSF cryptographic data.
- 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 current version and upgrade the firmware embedded in the TOE appliance. When installing updated firmware, digital signatures are used to authenticate the update to ensure it is the update intended and originated by the vendor.

6.7 TOE access

Whether connecting to the CLI locally or remotely (SSH), the TOE displays an advisory message when an administrator logs on. The message is configurable by TOE administrators.

The TOE locks local administrative sessions after an administrator configured period of inactivity. The inactivity period can be configured to be 1 to 3600 seconds with a default of 300 seconds. The TSF blanks the local console when it locks the session and requires the administrator to re-authenticate before allowing administrative access.

The TOE terminates remote administrative sessions after an administrator configured period of inactivity. The inactivity period can be configured to be 1 to 3600 seconds with a default of 300 seconds.

For client VPN connections, the TOE authenticates the connection if the session meets the session establishment rules by verifying (as configured):

- The client IP address is on the allowed list
- Connections are allowed during the current time of day
- Connections are allowed on the current date
- Connections are allowed during the current day of week
- Connections are allowed during the current day of month

If one of the session establishment rules above is not met the connection is rejected.

If the client VPN connection is authenticated and meets the session establishment rules, the TSF assigns the client a private IP address out of an administrator configured pool.

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.
- VPNGW12:FTA_SSL.3/VPN: The TOE monitors the client VPN activity within the Phase 2/Child SA. If no packets are sent for an administrator configured period of time, the TOE terminates the client VPN. The inactivity period can be configured to be 5 – 480 minutes.
- 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 locks local sessions that have been inactive for an administrator-configured period of time.
- NDcPP22e:FTA_TAB.1: The TOE displays an advisory warning banner regarding use of the TOE prior to establishing an administrator session. The administrator can configure the warning message displayed in the banner.
- VPNGW12:FTA_TSE.1: The TOE can deny establishment of a remote VPN client session based on location, time, and day.
- VPNGW12:FTA_VCM_EXT.1: The TOE assigns a private IP address out of an administrator configured pool to a VPN client upon a successful establishment of a session.

6.8 Trusted path/channels

The TSF utilizes SSH to provide the trusted path with protection from disclosure and modification for all remote administration sessions. Optionally, IPsec can be used to protect the SSH session.

The TSF uses IPsec to protect communications with the audit server, VPN peers, and the NTP server. The TSF utilizes X.509 certificates or PSKs to authenticate the non-TSF endpoint. The TOE can optionally use SSH to protect communication with the audit server.

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

- NDcPP22e:FTP_ITC.1: The TOE uses IPsec or SSH when exporting audit records to a third party syslog server and IPsec when communicating with an NTP server. The TOE also uses IPsec when communicating with IPsec clients and peers.

- VPNGW12:FTP_ITC.1/VPN: The TOE uses IPsec to provide a communication channel between itself and remote VPN gateway peers.
- NDcPP22e:FTP_TRP.1/Admin: The TOE uses SSH and optionally IPsec to provide a trusted path for remote management interfaces to protect the communication from disclosure and modification.