Aruba ClearPass Policy Manager 6.9 (NDcPP21) Security Target

Version 1.1
August 26, 2020

Prepared for:

Aruba, a Hewlett Packard Enterprise company

3333 Scott Blvd.
Santa Clara, CA 95054

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 ClearPass Policy Manager provided Aruba, a Hewlett Packard Enterprise company. The TOE is being evaluated as a Network Device.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In 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 – Aruba ClearPass Policy Manager 6.9 (NDcPP21) Security Target

ST Version – Version 1.1

ST Date – August 26, 2020

1.2 TOE Reference

TOE Identification – Aruba ClearPass Policy Manager version 6.9 running in one of the following appliances: C1000, C2000, C3000, or C3010.

TOE Developer – Aruba, a Hewlett Packard Enterprise company

Evaluation Sponsor – Aruba, a Hewlett Packard Enterprise company

1.3 TOE Overview

The Target of Evaluation (TOE) is Aruba ClearPass Policy Manager 6.9.

1.4 TOE Description

The Aruba ClearPass Policy Manager platform provides role- and device-based network access control for employees, contractors and guests across any wired, wireless and VPN infrastructure. ClearPass implements RADIUS services, as well as profiling, onboarding, guest access, and health checks facilitating centralized management of network access policies.

ClearPass provides user and device authentication based on 802.1X, non-802.1X and web portal access methods. Multiple authentication protocols like PEAP, EAP-FAST, EAP-TLS, and EAP-TTLS can be used concurrently to strengthen security in any environment. Attributes from multiple identity stores such as Microsoft Active Directory, LDAP-compliant directory, ODBC-compliant SQL database, token servers and internal databases can be used within a single policy for fine-grained control.

Additional information about the supported network access control capabilities can be found in the ClearPass Policy Manager data sheet (http://www.arubanetworks.com/pdf/products/DS_ClearPass_PolicyManager.pdf); however, for the purpose of evaluation, ClearPass will be treated as a network infrastructure device offering FIPS certified cryptographic functions, security auditing, secure administration, trusted updates, self-tests, and secure connections to other servers (e.g., to transmit audit records).

1.4.1 TOE Architecture

The ClearPass Policy Manager is available either as a hardware or virtual network appliance and is designed to support a wide range of network, wireless and security protocols to support a wide range of clients. However, the evaluation is limited to the hardware network appliances and the secure communication protocols specifically identified below.

There are five TOE appliance models designed to support different numbers of client devices. Each platform differs in CPU performance (e.g., number of cores), available memory, disk performance and storage capacity, and power consumption/supply.

Appliance Model	CPU
C1000	Intel Atom C2758 (Rangeley)
C2000	Intel Xeon E3-1240 v5 (Skylake)
C3000 (legacy only)	Intel Xeon E5-2620 v3 (Haswell)
C3010	Intel Xeon Gold 5118 (Skylake)

Table 1-1 TOE Models

While ClearPass Policy Manager products can be configured as a collection of devices operating in a cluster sharing a common security policy, the TOE configuration subject to this evaluation is limited to a single ClearPass Policy Manager device.

Each ClearPass Policy Manager device is a rack-mountable appliance with Intel Atom or Xeon CPUs running a version of CentOS 7.7 to host the applications designed to provide the network access control capabilities summarized above. ClearPass includes a version of Hewlett Packard Enterprise SSL crypto module that is used to perform cryptographic functions. This module is based on SafeLogic CryptoComply Version 2.1 and supports the implementations of IPsec using StrongSwan, TLS/HTTPS using Apache, and SSH using OpenSSH used to secure the communication channels (for remote administration, exporting audit events, and syncing with an NTP server).

1.4.1.1 Physical Boundaries

The physical boundaries of the TOE consist of ClearPass Policy Manager device running software version 6.9.

The ClearPass evaluated configuration includes one of the devices shown in Table 1-1 TOE Models.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by ClearPass:

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

1.4.1.2.1 Security audit

The TOE is designed to be able to generate logs for a wide range of security relevant events. The TOE can be configured to store the logs locally so they can be accessed by an administrator or alternately to send the logs to a designated syslog server.

1.4.1.2.2 Cryptographic support

The TOE includes an Aruba Linux 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, SSH, and TLS/HTTPS.

1.4.1.2.3 Identification and authentication

The TOE offers no TSF-mediated functions except display of a login banner until the administrator is identified and authenticated. The TOE authenticates administrative users accessing the TOE via the command-line interface (local serial console or SSH) or web interface (Web UI) in the same manner using its own password-based authentication mechanism. The TOE also supports public-key based authentication of users through the SSH-based CLI interface and supports certificate authentication for the Web UI.

The TOE supports certificate authentication for TLS and IPsec and supports pre-shared key authentication for IPsec connections. The TOE uses X.509v3 certificates and validates received authentication certificates. CRL and OCSP are supported for X509v3 certificate validation.

1.4.1.2.4 Security management

The TOE provides Command Line (CLI) commands (locally via a serial console or remotely via SSH) and a Web-based Graphical User Interface (Web GUI) to access the available functions to manage the TOE security functions. Security management commands are limited to authorized users (i.e., administrators) only after they have been correctly identified and authenticated. The security management functions are controlled through the use of Admin Privileges that can be assigned to TOE users.

1.4.1.2.5 Protection of the TSF

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

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

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

1.4.1.2.6 TOE access

The TOE can be configured to display an informative banner when an administrator establishes an interactive session and subsequently will enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated. The TOE can also reject authentication requests based on time of day, account status, location and role mapping.

1.4.1.2.7 Trusted path/channels

The TOE protects interactive communication with administrators using a console and SSHv2 for CLI access and TLS/HTTPS for Web UI access. In each case, both the integrity and disclosure protection is ensured via the secure protocol. If the negotiation of a secure session fails or if the user cannot be authenticated for remote administration, the attempted session will not be established.

The TOE protects communication with network peers, such as a syslog server or NTP server, using IPsec connections to prevent unintended disclosure or modification of logs.

1.4.2 TOE Documentation

The following administrator and user guidance are available:

- Common Criteria Configuration Guidance Aruba ClearPass Policy Manager 6.9, Version 4.1, August 2020 (**Admin Guide**)

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:

    collaborative Protection Profile for Network Devices, Version 2.1, 24 September 2018 with the following technical decisions:

    - TD0484
    - TD0483
    - TD0482
    - TD0481
    - TD0480
    - TD0478
    - TD0477
    - TD0475
    - TD0453
    - TD0451
    - TD0450
    - TD0447
    - TD0425
    - TD0424
    - TD0423
    - TD0412
    - TD0411
    - TD0410
    - TD0409
    - TD0408
    - TD0407
    - TD0402
    - TD0401
    - TD0400
    - TD0399
    - TD0398
    - TD0397
    - TD0396
    - TD0395

2.1 Conformance Rationale

The ST conforms to the NDcPP21. 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 NDcPP21 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP21 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP21 should be consulted if there is interest in that material.

In general, the NDcPP21 has defined Security Objectives appropriate for Network Device and as such are applicable to the ClearPass Policy Manager TOE.

3.1 Security Objectives for the Operational Environment

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

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

OE.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.

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.

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
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.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP21. The NDcPP21 defines the following extended requirements and since they are not redefined in this ST the NDcPP21 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP21:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
- NDcPP21:FCS_NTP_EXT.1: NTP Protocol
- NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
- NDcPP21:FCS_TLSS_EXT.2: TLS Server Protocol with mutual authentication
- NDcPP21:FIA_PMG_EXT.1: Password Management
- NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP21:FIA_X509_EXT.3: X.509 Certificate Requests
- NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP21:FPT_TST_EXT.1: TSF testing
- NDcPP21:FPT_TUD_EXT.1: Trusted update
- NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking

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 NDcPP21. The refinements and operations already performed in the NDcPP21 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP21 and any residual operations have been completed herein. Of particular note, the NDcPP21 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 NDcPP21 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP21 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP21 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by ClearPass Policy Manager TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP21:FAU_GEN.1: Audit Data Generation
	NDcPP21:FAU_GEN.2: User identity association
	NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP21:FCS_CKM.1: Cryptographic Key Generation
	NDcPP21:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP21:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP21:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP21:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP21:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP21:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP21: FCS_HTTPS_EXT.1: HTTPS Protocol
	NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
	NDcPP21:FCS_NTP_EXT.1: NTP Protocol
	NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
	NDcPP21:FCS_TLSS_EXT.2: TLS Server Protocol with mutual authentication
FIA: Identification and authentication	NDcPP21:FIA_AFL.1: Authentication Failure Management
	NDcPP21:FIA_PMG_EXT.1: Password Management
	NDcPP21:FIA_UAU.7: Protected Authentication Feedback
	NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP21:FIA_X509_EXT.3: X.509 Certificate Requests

FMT: Security management	NDcPP21:FMT_MOF.1/AutoUpdate: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/Functions: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/Services: Management of security functions behaviour
	NDcPP21:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP21:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP21:FMT_SMF.1: Specification of Management Functions
	NDcPP21:FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
	NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
	NDcPP21:FPT_TST_EXT.1: TSF testing
	NDcPP21:FPT_TUD_EXT.1: Trusted update
FTA: TOE access	NDcPP21:FTA_SSL.3: TSF-initiated Termination
	NDcPP21:FTA_SSL.4: User-initiated Termination
	NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP21:FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	NDcPP21:FTP_ITC.1: Inter-TSF trusted channel
	NDcPP21:FTP_TRP.1/Admin: Trusted Path

Table 5-1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP21:FAU_GEN.1)

NDcPP21:FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [***no other actions***];

d) Specifically defined auditable events listed in Table 5-2.

Table 5-2 Auditable Events

Requirement	Auditable Events	Additional Content
NDcPP21:FAU_GEN.1	None	None
NDcPP21:FAU_GEN.2	None	None

NDcPP21:FAU_STG_EXT.1	None	None
NDcPP21:FCS_CKM.1	None	None
NDcPP21:FCS_CKM.2	None	None
NDcPP21:FCS_CKM.4	None	None
NDcPP21:FCS_COP.1/DataEncryption	None	None
NDcPP21:FCS_COP.1/Hash	None	None
NDcPP21:FCS_COP.1/KeyedHash	None	None
NDcPP21:FCS_COP.1/SigGen	None	None
NDcPP21: FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Failure to establish a HTTPS Session.
NDcPP21:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
NDcPP21:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server
NDcPP21:FCS_RBG_EXT.1	None	None
NDcPP21:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP21:FCS_TLSS_EXT.2	Failure to establish a TLS Session.	Reason for failure.
NDcPP21:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_PMG_EXT.1	None	None
NDcPP21:FIA_UAU.7	None	None
NDcPP21:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
NDcPP21:FIA_X509_EXT.2	None	None
NDcPP21:FIA_X509_EXT.3	None	None
NDcPP21:FMT_MOF.1/AutoUpdate	None	None
NDcPP21:FMT_MOF.1/Functions	None	None
NDcPP21:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP21:FMT_MOF.1/Services	None	None
NDcPP21:FMT_MTD.1/CoreData	None	None
NDcPP21:FMT_MTD.1/CryptoKeys	None	None
NDcPP21:FMT_SMF.1	All management activities of TSF data.	None
NDcPP21:FMT_SMR.2	None	None
NDcPP21:FPT_APW_EXT.1	None	None
NDcPP21:FPT_SKP_EXT.1	None	None
NDcPP21: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	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).

	application note on FPT_STM_EXT.1)	
NDcPP21:FPT_TST_EXT.1	None	None
NDcPP21:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
NDcPP21:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
NDcPP21:FTA_SSL.4	The termination of an interactive session.	None
NDcPP21:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	None
NDcPP21:FTA_TAB.1	None	None
NDcPP21: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.
NDcPP21:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None

NDcPP21: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 5-2.

5.1.1.2 User identity association (NDcPP21:FAU_GEN.2)

NDcPP21: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 (NDcPP21:FAU_STG_EXT.1)

NDcPP21: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.

NDcPP21:FAU_STG_EXT.1.2

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

NDcPP21:FAU_STG_EXT.1.3

The TSF shall [***overwrite previous audit records according to the following rule: [audit records older than admin configured days (default value 7) are removed daily]***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP21:FCS_CKM.1)

NDcPP21: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] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4,***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1,***
- ***FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3***].

5.1.2.2 Cryptographic Key Establishment (NDcPP21:FCS_CKM.2)

NDcPP21: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" (TD0402 applied),***
- ***Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',***
- ***Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',***
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3].***

5.1.2.3 Cryptographic Key Destruction (NDcPP21:FCS_CKM.4)

NDcPP21:FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes, a new value of the key]***]

that meets the following: No Standard.

5.1.2.4 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP21:FCS_COP.1/DataEncryption)

NDcPP21:FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, CTR, GCM***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772***].

5.1.2.5 Cryptographic Operation (Hash Algorithm) (NDcPP21:FCS_COP.1/Hash)

NDcPP21: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***] that meet the following: ISO/IEC 10118-3:2004.

5.1.2.6 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP21:FCS_COP.1/KeyedHash)

NDcPP21: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 [***key size equal to digest size***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

5.1.2.7 Cryptographic Operation (Signature Generation and Verification) (NDcPP21:FCS_COP.1/SigGen)

NDcPP21: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 or greater],***
- ***Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]***]

that meet the following: [

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

5.1.2.8 HTTPS Protocol (NDcPP21: FCS_HTTPS_EXT.1)

NDcPP21:FCS_HTTPS_EXT.1.1

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

NDcPP21:FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

NDcPP21:FCS_HTTPS_EXT.1.3

If a peer certificate is presented, the TSF shall **[*not establish the connection*]** if the peer certificate is deemed invalid.

5.1.2.9 IPsec Protocol (NDcPP21:FCS_IPSEC_EXT.1)

NDcPP21:FCS_IPSEC_EXT.1.1

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

NDcPP21: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.

NDcPP21:FCS_IPSEC_EXT.1.3

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

NDcPP21: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 by RFC 3602)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384***] and [***AES-GCM-128***, ***AES-GCM-256 (specified in RFC 4106)***].

NDcPP21: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, [RFC 4304 for extended sequence numbers], and [RFC 4868 for hash functions],***
- ***IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23], and [RFC 4868 for hash functions]***].

NDcPP21: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)***].

NDcPP21:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [

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

NDcPP21:FCS_IPSEC_EXT.1.8

The TSF shall ensure that [

- ***IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [o length of time, where the time values can be configured within [8] hours],***
- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [o length of time, where the time values can be configured within [8] hours]***].

NDcPP21: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, 256, or 384***] bits.

NDcPP21:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv1, IKEv2***] exchanges of length [

- ***according to the security strength associated with the negotiated Diffie-Hellman group;***
- ***at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

NDcPP21:FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s) [***14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP)***].

NDcPP21: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.

NDcPP21: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***].

NDcPP21: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)***] and [***no other reference identifier type***].

5.1.2.10 NTP Protocol (NDcPP21:FCS_NTP_EXT.1)

NDcPP21:FCS_NTP_EXT.1.1

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

NDcPP21:FCS_NTP_EXT.1.2
The TSF shall update its system time using [
- ***Authentication using [SHA1] as the message digest algorithm(s);***
- ***[IPsec] to provide trusted communication between itself and an NTP time source.***].

NDcPP21:FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses

NDcPP21:FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources.

5.1.2.11 Random Bit Generation (NDcPP21:FCS_RBG_EXT.1)

NDcPP21: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)***].

NDcPP21:FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[one] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

5.1.2.12 SSH Server Protocol (NDcPP21:FCS_SSHS_EXT.1)

NDcPP21:FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5656, 6668***]. (TD0398 applied)

NDcPP21:FCS_SSHS_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***password-based***].

NDcPP21:FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than [***256K***] bytes in an SSH transport connection are dropped.

NDcPP21: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***].

NDcPP21:FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, ecdsa-sha2-nistp256***] as its public key algorithm(s) and rejects all other public key algorithms. (TD0424 applied)

NDcPP21:FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha2-256, hmac-sha2-512, implicit***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

NDcPP21:FCS_SSHS_EXT.1.7
The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***no other methods***] are the only allowed key exchange methods used for the SSH protocol.

NDcPP21: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. (TD0475 applied)

5.1.2.13 TLS Server Protocol with mutual authentication (NDcPP21:FCS_TLSS_EXT.2)

NDcPP21:FCS_TLSS_EXT.2.1

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

- ***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,***
- ***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***].

NDcPP21:FCS_TLSS_EXT.2.2

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

NDcPP21:FCS_TLSS_EXT.2.3

The TSF shall [***perform RSA key establishment with key size [2048 bits, 3072 bits, 4096 bits], generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1] and no other curves, generate Diffie-Hellman parameters of size [2048 bits, 3072 bits]***].

NDcPP21:FCS_TLSS_EXT.2.4

The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

NDcPP21:FCS_TLSS_EXT.2.5

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [***Not implement any administrator override mechanism***].

NDcPP21:FCS_TLSS_EXT.2.6

The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication Failure Management (NDcPP21:FIA_AFL.1)

NDcPP21:FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within [***0 to 100***] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password. (TD0408 applied)

NDcPP21:FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until [an explicit action to unlock] is taken by an Administrator (CLI and Web UI); prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed (CLI Only)***]. (TD0408 applied)

5.1.3.2 Password Management (NDcPP21:FIA_PMG_EXT.1)

NDcPP21:FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')'***];

b) Minimum password length shall be configurable to between [***6***] and [***100***] characters.

5.1.3.3 Protected Authentication Feedback (NDcPP21:FIA_UAU.7)

NDcPP21:FIA_UAU.7.1

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

5.1.3.4 Password-based Authentication Mechanism (NDcPP21:FIA_UAU_EXT.2)

NDcPP21:FIA_UAU_EXT.2.1

The TSF shall provide a local [***password-based, SSH public key-based, certificate-based***] authentication mechanism to perform local administrative user authentication. (TD0408 applied)

5.1.3.5 User Identification and Authentication (NDcPP21:FIA_UIA_EXT.1)

NDcPP21: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***].

NDcPP21:FIA_UIA_EXT.1.2

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

5.1.3.6 X.509 Certificate Validation (NDcPP21:FIA_X509_EXT.1/Rev)

NDcPP21:FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

NDcPP21:FIA_X509_EXT.1.2/Rev

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

5.1.3.7 X.509 Certificate Authentication (NDcPP21:FIA_X509_EXT.2)

NDcPP21:FIA_X509_EXT.2.1

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

NDcPP21:FIA_X509_EXT.2.2

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

5.1.3.8 X.509 Certificate Requests (NDcPP21:FIA_X509_EXT.3)

NDcPP21:FIA_X509_EXT.3.1

The TSF shall generate a Certification Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***]. (TD0333 applied)

NDcPP21:FIA_X509_EXT.3.2

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

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (NDcPP21:FMT_MOF.1/AutoUpdate)

NDcPP21:FMT_MOF.1.1/AutoUpdate

The TSF shall restrict the ability to [***enable, disable***] the functions [***automatic checking for updates***] to Security Administrators.

5.1.4.2 Management of security functions behaviour (NDcPP21:FMT_MOF.1/Functions)

NDcPP21:FMT_MOF.1.1/Functions

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

5.1.4.3 Management of security functions behaviour (NDcPP21:FMT_MOF.1/ManualUpdate)

NDcPP21:FMT_MOF.1.1/ManualUpdate

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

5.1.4.4 Management of security functions behaviour (NDcPP21:FMT_MOF.1/Services)

NDcPP21:FMT_MOF.1.1/Services
The TSF shall restrict the ability to enable and disable start and stop services to Security Administrators.

5.1.4.5 Management of TSF Data (NDcPP21:FMT_MTD.1/CoreData)

NDcPP21:FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the TSF data to Security Administrators.

5.1.4.6 Management of TSF Data (NDcPP21:FMT_MTD.1/CryptoKeys)

NDcPP21:FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.1.4.7 Specification of Management Functions (NDcPP21:FMT_SMF.1)

NDcPP21: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,***
- ***Ability to modify the behavior of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full,***
- ***Ability to manage the cryptographic keys,***
- ***Ability to enable or disable automatic checking for updates or automatic updates;***
- ***Ability to re-enable an Administrator account,***
- ***Ability to set the time which is used for time-stamps;***
- ***Ability to configure NTP***].

5.1.4.8 Restrictions on Security Roles (NDcPP21:FMT_SMR.2)

NDcPP21:FMT_SMR.2.1
The TSF shall maintain the roles:
- Security Administrator.

NDcPP21:FMT_SMR.2.2
The TSF shall be able to associate users with roles.

NDcPP21: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 Protection of the TSF (FPT)

5.1.5.1 Protection of Administrator Passwords (NDcPP21:FPT_APW_EXT.1)

NDcPP21:FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form. (TD0483 applied)

NDcPP21:FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords. (TD0483 applied)

5.1.5.2 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP21:FPT_SKP_EXT.1)

NDcPP21:FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.5.3 Reliable Time Stamps (NDcPP21:FPT_STM_EXT.1)

NDcPP21:FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

NDcPP21:FPT_STM_EXT.1.2
The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server***].

5.1.5.4 TSF testing (NDcPP21:FPT_TST_EXT.1)

NDcPP21: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: [***cryptograph library self-tests and TOE integrity tests***].

5.1.5.5 Trusted update (NDcPP21:FPT_TUD_EXT.1)

NDcPP21: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***].

NDcPP21:FPT_TUD_EXT.1.2
The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***support automatic checking for updates***].

NDcPP21:FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature mechanism***] prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 TSF-initiated Termination (NDcPP21:FTA_SSL.3)

NDcPP21:FTA_SSL.3.1
The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.6.2 User-initiated Termination (NDcPP21:FTA_SSL.4)

NDcPP21:FTA_SSL.4.1
The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.1.6.3 TSF-initiated Session Locking (NDcPP21:FTA_SSL_EXT.1)

NDcPP21:FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [***- terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.6.4 Default TOE Access Banners (NDcPP21:FTA_TAB.1)

NDcPP21: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 Trusted path/channels (FTP)

5.1.7.1 Inter-TSF trusted channel (NDcPP21:FTP_ITC.1)

NDcPP21:FTP_ITC.1.1
The TSF shall be capable of using [***IPsec***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[NTP Server]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

NDcPP21:FTP_ITC.1.2
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

NDcPP21:FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [***syslog, NTP***].

5.1.7.2 Trusted Path (NDcPP21:FTP_TRP.1/Admin)

NDcPP21:FTP_TRP.1.1/Admin
The TSF shall be capable of using [***SSH, TLS, HTTPS***] to provide a communication path between itself and authorized remote Administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

NDcPP21:FTP_TRP.1.2/Admin
The TSF shall permit remote Administrators to initiate communication via the trusted path.

NDcPP21: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

Table 5-3 **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
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TOE generates audit records for start-up and shutdown of the TOE, all administrator actions, and for an unspecified level of audit (see Table 5-2 Auditable Events for specific events). Audit records include date and time of the event, type of event, user identity that caused the event to be generated, and the outcome of the event. For any auditable events related to cryptographic key operations, the key or certificate name is logged. The TOE maintains local audit logs that are only accessible for View access by TOE administrators after logging in.

There are three locations in the Web UI where audit records are stored and can be viewed: Access Tracker, Audit Viewer, and Event Viewer.

By default, the Access Tracker and Audit Viewer store the logs for 7 days after which time they will be deleted automatically. The automatic clean up period can be configured by the administrator to be longer or shorter as may be necessary for a given deployment. The Audit Viewer storage can be configured via the cleanup parameter "Old Audit Records Cleanup Interval". The Access Tracker storage can be configured via the cleanup parameter "Cleanup interval for Session Log details in database". The Event Viewer records are stored for seven (7) days after which time they will be deleted automatically. There is no user configurable setting to modify the Event Viewer log storage.

The number and size of log files may be specified based on observed logging levels. The default number of log files is 12 and the default size of each log file is 50MB. The specific capacity of the audit storage is dependent on the disk drive capability of the TOE. The default disk capacity has been designed so that in a typical deployment the available space will not be exhausted within the default retention periods. Disk usage settings will notify the administrator if the system is running with low disk space.

The TOE can also be configured to send audit records to a trusted third party SYSLOG server in the operational environment.

The TOE is a standalone TOE that stores audit data locally and transfers audit data to an external syslog server periodically. ClearPass does not transfer syslog messages in real time. Messages are queued to a syslog buffer that then transfers all messages to the syslog server every 120 seconds. This value may be reduced to a minimum of every 30 seconds, but will default to every 120 seconds.

The Security audit function satisfies the following security functional requirements:

- NDcPP21:FAU_GEN.1: The TOE generates audit events for the not specified level of audit. A syslog server in the environment is relied on to store audit records generated by the TOE.
- NDcPP21: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.
- NDcPP21: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. Also, any audit records older than an administrator configured period (default 7 days) are deleted daily.

6.2 Cryptographic support

The TOE includes the Hewlett Packard Enterprise SSL crypto module, Version 2.1 that provides supporting cryptographic functions. The evaluated configuration requires that the TOE be configured in FIPS mode to ensure CAVP certified functions are used.

The following functions have been CAVP certified in accordance with the identified standards.

Table 6-1 Cryptographic Functions

Requirements	Functions	Cert
	Cryptographic key generation	
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater	C1773
	ECC schemes using 'NIST curves' P-256 and P-384	C1773
	FFC schemes using cryptographic key sizes of 2048-bit or greater (DSA)	C1773
	Cryptographic key establishment	
FCS_CKM.2	RSA-based key establishment schemes	Vendor Affirmed
	Elliptic curve-based key establishment schemes	C1773
	Finite field-based key establishment schemes (KAS ECC/FFC)	C1773
	Encryption/Decryption	
FCS_COP.1/Data Encryption	AES CBC (128 and 256 bits)	C1773
	AES GCM (128 and 256 bits)	C1773
	AES CTR (128 and 256 bits)	C1773
	Cryptographic signature services	
FCS_COP.1/SigGen	RSA Digital Signature Algorithm (rDSA) (2048 bits or greater)	C1773
	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve size of 256 or 384	C1773
	Cryptographic hashing	
FCS_COP.1/Hash	SHA-1/256/384/512 (digest sizes 160, 256, 384 bits and 512 bits)	C1773
	Keyed-hash message authentication	
FCS_COP.1/KeyedHash	HMAC-SHA-1 (block size 512 bits, key and digest size 160 bits) HMAC-SHA-256 (block size 512 bits, key and digest size 256 bits) HMAC-SHA-384 (block size 1024 bits, key and digest size 384 bits), HMAC-SHA-512 (block size 1024 bits, key and digest size 512 bits)	C1773
	Random bit generation	
FCS_RBG_EXT.1	CTR_DRBG (AES) with S/W based noise source	C1773

The TOE generally fulfills all of the NIST SP 800-56A and Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1” requirements without extensions. The TOE does not perform any operations marked as “shall not” or “should not” and performs all operations marked as “shall” or "should". For finite-field based key establishment, the TOE implements the following sections of SP 800-56A: 5.6 and all subsections. For RSA key establishment, the TOE implements Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1". The TOE also supports key establishment using Diffie-Hellman group 14 that meets Section 3 of RFC 3526.

Security Function	Communication Type	Key Establishment Methods
Administration	TLS	RSA Schemes ECC Schemes FFC Schemes DH-14
Administration	SSH	ECC Schemes FFC Schemes DH-14
Trusted Channels for Syslog, NTP	IPsec	ECC Schemes FFC Schemes DH-14

Table 6-2 Key Exchange Methods used by TOE Services

The TOE uses a software-based random bit generator that complies with AES-256 CTR_DRBG when operating in the FIPS mode. AES-256 is used in conjunction with a minimum of 384 bits of entropy from jitter entropy.

Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. Note that zeroization occurs as follows: 1) when deleted from the encrypted drive, the previous value is overwritten once with zeroes; 2) when added or changed on the encrypted drive, any old value is overwritten completely with the new value; and, 3) the zeroization of values in RAM is achieved by overwriting once with zeroes. All operations on the encrypted drive and RAM utilize standard file system APIs or memory management APIs.

The following Critical Security Parameters and keys are subject to key destruction:

- Server Private Keys (RSA or ECDSA) - stored on encrypted drive and overwritten when replaced
- SSH Authentication Keys - stored on encrypted drive and overwritten when replaced or cleared when removed
- SSH Session Keys - stored in RAM and overwritten when the session terminates
- SSH KDF Internal State - stored in RAM and overwritten when the session terminates
- SSH Shared Secret Key - stored in RAM and overwritten when the session terminates
- TLS Pre-Master Secret - stored in RAM and overwritten when the session terminates
- TLS Master Secret - stored in RAM and overwritten when the session terminates
- TLS PRF Internal State - stored in RAM and overwritten when the session terminates
- TLS Session Key - stored in RAM and overwritten when the session terminates
- TLS Authentication Key for HMAC-SHA-X - stored in RAM and overwritten when the session terminates
- RNG Seed Material - stored in RAM and overwritten when used
- RNG Internal State - stored in RAM and overwritten when shutdown
- IKE Session Encryption Key - stored in RAM and overwritten when the session terminates
- IKE Session Authentication Key - stored in RAM and overwritten when the session terminates
- IPsec Encryption Key - stored in RAM and overwritten when the session terminates
- IPsec Authentication Key - stored in RAM and overwritten when the session terminates
- Passwords - stored on encrypted drive and overwritten when changed or cleared when removed

These supporting cryptographic functions are included to support IPsec (compliant with RFC 4301), SSHv2 (compliant with RFCs 4251, 4252, 4253, and 4254), and TLSv1.2 (compliant with RFC 5246) secure communication protocols.

The TOE supports IPsec for both transport and tunnel mode. For ESP encryption and the encrypted payload in IKEv1 the TOE supports 128 and 256-bit AES-CBC. For ESP encryption and the encrypted payload in IKEv2, the TOE supports 128 and 256-bit AES-CBC or 128 and 256-bit AES_GCM. Similarly, HMAC-SHA1, HMAC-SHA-256 and HMAC-SHA384 are supported for keyed hashing. Diffie-Hellman (DH) Groups 14, 19, and 20 are supported for both IKEv1 and IKEv2 as are RSA and ECDSA certificates and pre-shared key IPsec authentication. The TOE selects the DH group by selecting the largest group configured by an administrator that is offered by the VPN gateway. Note that aggressive mode is not used with IKEv1, only main mode is supported. When configuring ciphers, there is only one setting that applies to both phase 1 and phase 2, this ensures that the IKE and ESP ciphers are the same and hence have the same security strength.

IPsec connections can be configured by identifying a TOE interface and peer IP address and IPsec-specific connection parameters: tunnel/transport mode, IKE version, encryption and hash algorithms, Diffie-hellman group, and authentication type. IKEv1 Phase 1 SA and IKEv2 SA lifetime limits can be configured to be up to 24 hours by a Security Administrator. Similarly, IKEv1 Phase 2 SA and IKEv2 Child SA lifetime limits can be configured up to 8 hours by a Security Administrator. After SAs are established as part of a connection, each SA is renegotiated and re-established each time its configured lifetime is reached. When an IPsec connection is configured, the administrator can define the DN for the peer. When the connection is made, the configured DN is compared against that in the peer certificate and the connection succeeds only if they match exactly.

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, 19, and 20, respectively). When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life of a specific IPsec SA is less than 1 in 2^{112}, 2^{128}, or 2^{192}, corresponding to the respective DH group. For IKEv2, the nonces used in the IKE exchanges are generated by the TOE's random bit generator with lengths of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash.

The TOE supports the definition of "IPsec Traffic Selector Rules". The default behavior for IPsec rules is to encrypt all traffic between the TOE and a VPN peer. Traffic can be separated on a per-port and/or per-protocol level for encrypt, bypass, or drop actions. When implementing IKEv1, only one (1) rule of each type may be created. When implementing IKEv2, a maximum of ten (10) rules may be created for each IPsec tunnel.

The actions associated with each rule type are:

- **Encrypt Rules** - All packets matching these rules will be encrypted through the IPsec tunnel. When no subordinate actions are specified, this is the default for all traffic between hosts.
- **Bypass Rules** - All packets matching these rules will bypass the IPsec tunnel and flow to the remote peer outside of the VPN. This is commonly known as traffic "in the clear", even though it may already be encrypted. When using bypass rules, both peers must be configured to bypass the selected traffic or the remote end will not appropriately process the packets.
- **Drop Rules** - All packets matching these rules will be dropped.
- **Final Rule** - An implicit rule is created with all IPsec traffic selection that will drop any traffic not processed. This rule will create a behavior where all traffic that should be encrypted or dropped between peers will always be blocked when the VPN is inactive. Bypass traffic is unaffected by tunnel status.

The defined IPsec rules are processed using both order and specificity. Order is established beginning by rule position starting with the first rule and descending within a rule group. Specificity is established based on the exactness of a rule to match against. Rules with specific ports and protocols will be evaluated prior to more general rules that apply to all ports or protocols prior to rules that catch "any" traffic.

The TOE supports SSHv2 with aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, and aes256-gcm@openssh.com encryption algorithms, in conjunction with HMAC-SHA-1, HMAC-SHA2-256 and HMAC-SHA2-512 for data integrity and the following key exchange methods: diffie-hellman-group14-sha1 and ecdh-sha2-nistp256. Note: When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC.

The TOE's implementation of SSHv2 supports both public-key and password-based authentication; and packets are limited to 256K bytes. SSH public key authentication supports the ssh-rsa and ecdsa-sha2-nistp256 algorithms.

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 (256K bytes) the packet will be dropped. Also, once an SSH session is established, the TOE starts a timer and keeps track of data exchanged. Once either 128 MB of data is transferred or one hour elapses, the TOE issues a rekey message causing new keys to be exchanged between the TOE and the SSH client and the timer and data counters are reset.

The TOE supports TLSv1.2 with AES (CBC and GCM) 128 or 256-bit ciphers, in conjunction with SHA-1, SHA-256, and SHA-384 using RSA and ECDSA for authentication. Any other SSL/TLS versions are not supported by the TOE and such connection attempts will be rejected. The TOE performs key establishment, depending on the TLS cipher suite that is negotiated, using RSA with key sizes of 2048, 3072 or 4096 bits, ECDSA with secp256r1 or secp384r1 NIST curves, or using Diffie-Hellman with 2048 or 3072 bits. The SAN or CN in the certificate presented by the peer must match the expected identifier or the TOE will not establish the TLS connection. The TOE acts as a TLS server with mutual authentication and requires no additional configuration to support the evaluated ciphersuites listed in the NDcPP21:FCS_TLSS_EXT.2 requirement.

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP21:FCS_CKM.1: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_CKM.2: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_CKM.4: See “Critical Security Parameters and keys” list above.
- NDcPP21:FCS_COP.1/DataEncryption: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_COP.1/Hash: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_COP.1/KeyedHash: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_COP.1/SigGen: See Table 6-1 Cryptographic Functions above.
- NDcPP21:FCS_HTTPS_EXT.1: The TOE implements HTTPS using TLS and compliant with RFC 2818. Note that the TOE requires the peer to initiate the connection and the TOE can be configured to require mutual authentication and when so configured requires a valid certificate to be provided by the peer. The TOE will not establish a connection when an invalid certificate is presented.
- NDcPP21:FCS_IPSEC_EXT.1: The TOE supports IPsec to protect communication when exporting audit records as indicated above or when synching to an NTP server.
- NDcPP21:FCS_NTP_EXT.1: The TOE supports NTPv4, while rejecting all broadcast and multicast time updates. The TOE can authenticate an NTP server using a SHA1 key or can utilize NTP within an authenticated IPsec tunnel. The TOE can be configured to identify as many as 5 NTP servers from which time update are accepted.
- NDcPP21:FCS_RBG_EXT.1: See Table 6-1 Cryptographic Functions above. The TOE uses one software based noise source - Jitter Entropy daemon.
- NDcPP21:FCS_SSHS_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- NDcPP21:FCS_TLSS_EXT.2: The TOE supports TLS sessions in conjunction with HTTPS for web based administrator access. The TOE TLS server supports the cipher suites listed in NDcPP21:FCS_TLSS_EXT.2.1 for web based administrator access. For web based administrator access the TOE performs the following:
  - RSA key establishment with key size 2048 bits, 3072 bits, 4096 bits,
  - generates EC Diffie-Hellman parameters over NIST curves secp256r1, secp384r1

– generates Diffie-Hellman parameters of size 2048 bits, 3072 bits.

6.3 Identification and authentication

The TOE defines administrative users in terms of:

- User identity,
- User name,
- Password, and
- Admin Privileges.

Specific privileges are associated with privilege levels and serve to determine the functions the associated administrator can perform.

The TOE authenticates administrative users accessing the TOE via the command-line interface (local serial console or SSH) or web interface (Web UI) in the same manner using its own password-based authentication mechanism. The TOE also supports public-key based authentication of users through the SSH-based CLI interface and supports certificate authentication for the Web UI. In order for an administrative user to access the TOE (i.e., to perform any functions except to see a configured login banner or to access network access control services, including processing RADIUS, Web, and other authentication requests from external entities), an administrative user account must be created for the user with an assigned privilege level.

The TOE password authentication mechanism enforces password composition rules. A minimum password length can be configured to 15 characters. Passwords can generally contain alphabetic (upper or lower case) characters, numeric characters, and special characters such as any of ***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')'*** and they are case-sensitive. The TOE supports the configuration of password composition policies such as:

- No password complexity requirement;
- At least one uppercase and one lowercase letter;
- At least one digit;
- At least one letter and one digit;
- At least one of each: uppercase letter, lowercase letter, digit;
- At least one symbol; and
- At least one of each: uppercase letter, lowercase letter, digit, and symbol.

Additionally, disallowed characters and words can be defined along with even more checks such as disallowing repeating character four times or containing the user identity either forward or backwards. All of the configured policies are enforced whenever a user changes their password.

When authentication fails, the TOE increments a per-user counter. The per-user counter is reset to 0 upon successful authentication. If the per-user counter reaches the configured limit, the account is locked. For SSH-based CLI logins the per-user counter is reset to 0 when the account is explicitly unlocked or after the configured period, whichever occurs first. If the configured authentication threshold is exceeded on the Web UI, the account is locked out until an administrator resets the account to re-enable Web UI login for that account. Accounts are never locked out on the local console.

When authentication succeeds (regardless of interface), the TOE looks up the user's defined privilege level, assigns that to the user's session, and presents the user with a command prompt or interface. At this point the user has successfully logged on and can perform their authorized functions.

When configuring IPsec connections, both certificate- and pre-shared-key based authentication are supported. In the case of pre-shared keys, the administrator types in and confirms the pre-shared key. The pre-shared key can be up to 128 characters in length (e.g., including 22 characters). Certificates are also utilized for authentication when establishing TLS connections. In each case, when initiating a connection, the TOE presents a Security Administrator configured certificate.

The TOE uses X.509v3 certificates for IPsec and TLS connections. During connection establishment, the TOE validates received authentication certificates. If the certificate appears to be valid (e.g., is properly constructed and can be decoded), the TOE then validates that it can construct certificate path from the certificate through any intermediary CAs to a configured trusted root CA. If the path can be constructed, the validity date and CA flag is checked in each CA certificate. If all of those checks succeed, the TOE finally checks the revocation status using OCSP or a configured CRL of all certificates in the path. The TOE will reject any certificate for which it cannot determine validity and will reject the connection attempt.

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

- NDcPP21:FIA_AFL.1: For Web UI login attempt, when the failure limit is reached, the applicable administrator account is locked until an explicit unlock operation is taken by a local administrator. For SSH-based CLI login attempts, when the failure limit is reached the applicable administrator account is locked until either an explicit unlock operation is taken by a local administrator or a configurable period of time elapses. The accounts are never locked when used to access the local console.

  Note that an administrator account is defined for either the use of the WebUI or of the SSH-Based CLI interface.

- NDcPP21:FIA_PMG_EXT.1: The TOE supports passwords comprising upper and lower case alphabetic characters, numbers, and a set of special characters identified above. The TOE also allows administrator to define a minimum password length of between 6 and 100 characters.

- NDcPP21:FIA_UAU.7: The TOE does not echo passwords as they are entered; passwords are not echoed on the console or SSH interfaces and '.' characters are echoed on the Web UI when entering passwords.

- NDcPP21:FIA_UAU_EXT.2/NDcPP21:FIA_UIA_EXT.1: The TOE offers no TSF-mediated functions except display of a login banner until the user is identified and authenticated The TOE provides a password-based authentication mechanism, as well as public-key authentication for SSH and supports certificate authentication for the Web UI.

- NDcPP21:FIA_X509_EXT.1/Rev: CRL and OCSP are supported for X509v3 certificate validation

- NDcPP21:FIA_X509_EXT.2: When configured for OCSP or CRLs are configured for the applicable certificates, the TOE will reject connections if the revocation status cannot be determined. When configured for CRL, but a CRL is not configured for the applicable certificate, the connection will be accepted.

- NDcPP21:FIA_X509_EXT.3: The TOE generates certificate requests and validates the CA used to sign the certificates.

6.4 Security management

The TOE defines an administrator role that can be assigned more granular privileges via defined privilege levels. Each time a new administrative user is defined a user identifier, username, password, and privilege level must be assigned. There are a number of pre-defined privilege levels (e.g., Super Administrator, Network Administrator) while additional privilege levels can be defined by the TOE user as may be needed for a specific deployment.

The TOE administrative interfaces consist of network-based interfaces and a serial terminal-based interface. A command-line interface (CLI) can be accessed over the network using SSH or locally using the serial interface. The Web UI can be accessed using a web browser via TLS/HTTPS. The Web UI is the primary administrative interface, while many of the administrator commands are also available via the CLI.

Once authenticated (none of these functions are available to any user before being identified and authenticated), authorized administrators have access to the following security functions:

Using the Web UI:

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to configure a login banner;
- Ability to configure the IPsec functionality;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to enable and disable the automatic checking for updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1 including the ability to re-enable an Administrator account;
- Ability to start and stop services;
- Ability to configure audit behavior including configuring the TOE for the transmission of audit data to an external IT entity;
- Ability to manage the cryptographic keys;
- Ability to set the time which is used for time-stamps; and
- Ability to configure NTP.

Using the CLI:

- Ability to administer the TOE locally and remotely;
- Ability to configure the authentication failure parameters for FIA_AFL.1 including the ability to re-enable an Administrator account (for SSH);
- Ability to start and stop services.

The Security management function satisfies the following security functional requirements:

- NDcPP21:FMT_MOF.1/AutoUpdate: The TOE has the ability to enable and disable its automatic checking for updates.
- NDcPP21:FMT_MOF.1/Functions: The TOE allows administrators to configure the transmission of audit data to an external audit server.
- NDcPP21:FMT_MOF.1/ManualUpdate: Administrators can instruct the TOE to perform a product update.
- NDcPP21:FMT_MOF.1/Services: Administrators can start and stop services.
- NDcPP21:FMT_MTD.1/CoreData: The TOE restricts the access to manage TSF data that can affect the security functions of the TOE to authorized administrators.
- NDcPP21:FMT_MTD.1.1/CryptoKeys: The TOE restricts the ability to manage cryptographic keys to authorized administrators.
- NDcPP21:FMT_SMF.1: The TOE provides administrative interfaces to perform the functions identified above.
- NDcPP21:FMT_SMR.2: The TOE maintains administrative user roles.

6.5 Protection of the TSF

The TOE is an appliance and as such is designed to work independently of other components to a large extent. Secure communication with third-party trusted peers is addressed in section 6.8.

While the administrative interface is function rich, the TOE is designed specifically to not provide access to locally stored passwords and also, while cryptographic keys can be entered, the TOE does not disclose any cryptographic keys stored in the TOE. The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The TOE can be configured to periodically synchronize its clock with a time server, but the TOE can only ensure its own reliability and not that of an external time mechanism. The TOE also implements the timing elements through timeout functionality due to inactivity for terminating both local and remote sessions. Note that the clock is used primarily to provide timestamp for audit records, but is also used to supporting timing elements of cryptographic functions, certificate validity checks, session timeouts, and unlocking of administrator accounts locked as a result of authentication failure.

The TOE includes a number of built in diagnostic tests that are run during start-up to determine whether the TOE is operating properly. An administrator can configure the TOE to reboot or to stop, with errors displayed, when an error is encountered. When configured, the power-on self-tests comply with the FIPS 140-2 requirements for self-testing. The module performs Cryptographic algorithm known answer tests, firmware integrity tests using RSA signature verification and conditional self-tests for PRNG, Pair-wise consistency tests on generation of RSA keys, and a Firmware load test (RSA signature verification). Upon failing any of its FIPS mode power-on self-tests, the TOE will refuse to boot.

The TOE supports updating the TOE software using the Web UI. From the Web UI, an administrator can identify available updates and upgrades, download, and install or re-install them. Subsequently updates and upgrades would be identified as ‘installed’ or ‘install error’ indicating there was a problem with the installation. If the update server is not accessible, the administrator can also import updates. Of course, this requires that the administrator has access to the update (e.g., previous download, access update server from an alternate machine) and can import it directly into the TOE.

Signing and verifying the update/upgrade images uses a cryptographic digest function. A 2048-bit RSA keypair (self-signed) is generated and the binary image is signed using the private key. The public key is shipped with the TOE and is used for Verification of the signed.tar file. The tar file contains the signature and binary image (zip of binary + metafile). Once the tar file is extracted the TOE verifies whether the signature of the binary image and the extracted signature match. If it matches, verification is successful.

The TOE generates time stamps to support the auditing function.

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

- NDcPP21:FPT_APW_EXT.1: The TOE does not offer any functions that will disclose to any user a plain text password. Furthermore, locally defined passwords are not stored in plaintext form; they are stored hashed with PBKDF2 (1,000 iterations).
- NDcPP21:FPT_SKP_EXT.1: The TOE does not offer any functions that will disclose to any users a stored cryptographic key. Keys are generated during system bootstrapping and not exposed to users or administrators.
- NDcPP21:FPT_STM_EXT.1: The TOE generates time stamps for use in audit records, cryptographic functions, certificate validity checks, session timeouts, and unlocking of administrator accounts locked as a result of authentication failure.
- NDcPP21:FPT_TST_EXT.1: The TOE includes a number of power-on diagnostics that will serve to ensure the TOE is functioning properly. The tests include ensure memory can be accessed as expected, to ensure that software checksums are correct, and also to test the presence and function of plugged devices.
- NDcPP21:FPT_TUD_EXT.1: The TOE provides functions to query the version and upgrade the software embedded in the TOE appliance. When installing updated software, digital signatures are used to authenticate the update to ensure it is the update intended and originated by Aruba. The TOE supports automatic checking for updates through the HPE Passport system. The TOE obtains credentials for the HPE Passport system from an administrator and uses those credentials to authenticate to the HPE Passport system to check for newer versions of the TOE that may be available. Upon detecting that a newer version of the code is available the TOE informs the administrator through a message presented on the software updates page of the Web UI. Instructions for accessing the HPE Passport system are provided in the Admin Guide.

6.6 TOE access

The TOE is configured to display an administrator-configured login banner before authentication. In all cases (console, SSH, and web interface), the login banner is presented before an administrative user session is established.

The TOE is configured by an administrator to set a session timeout. A session (local console or remote SSH or Web/HTTPS) that is inactive (i.e., no commands issuing from the remote client) for the defined timeout value will be terminated. Upon exceeding the session timeout, the TOE logs the user off.

The user will be required to login in after any session has been terminated due to inactivity or after voluntary termination. Of course, administrators can logout of local or remote sessions at any time.

The TOE access function satisfies the following security functional requirements:

- NDcPP21:FTA_SSL.3: The TOE terminates remote sessions that have been inactive for an administrator-configured period of time.
- NDcPP21:FTA_SSL.4: The TOE provides the function to logout (i.e., terminate) both local and remote user sessions as directed by the user.
- NDcPP21:FTA_SSL_EXT.1: The TOE terminates local sessions that have been inactive for an administrator-configured period of time.
- NDcPP21:FTA_TAB.1: The TOE is configured to display administrator-defined advisory banners when administrators successfully establish interactive sessions with the TOE.

6.7 Trusted path/channels

The TOE provides a trusted path for its remote administrative users accessing the TOE via the Ethernet ports provided on the TOE using either a command line interface using SSH or Web-based graphical user interface using TLS/HTTPS. Local console access via a serial port is also supported for command line access. However, this access is protected by physical protection of the serial interface along with the TOE itself.

When an administrator attempts to connect to the TOE remotely, the TOE attempts to negotiate a session. If the session cannot be negotiated, the connection is dropped. When negotiating a TLS/HTTPS or SSH session, the TOE and the client application (SSH client or web browser) used by the administrator will negotiate the most secure algorithms available at both ends to protect that session. The available algorithms are identified in section 6.3 above.

Remote connections to trusted third party syslog servers are supported for exporting audit records. Communication with those external audit servers is protected using IPsec as specified in section 6.3.

The TOE can sync to an external NTP server over a protected IPsec tunnel as specified in section 6.3.

In all cases, the endpoints are assured by virtue of the certificates installed, trusted, and reviewable when connecting and by virtue of user authentication.

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

- NDcPP21:FTP_ITC.1: In the evaluated configuration, the TOE must be configured to use IPsec to ensure that any exported audit records are sent only to the configured server and are not subject to inappropriate disclosure or modification. IPsec is also used to protect communication to sync to an external NTP server.
- NDcPP21:FTP_TRP.1/Admin: The TOE uses SSH and TLS/HTTPS to provide a trusted path for remote management interfaces to protect the communication from disclosure and modification.