HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0 Security Target

Version 1.0
10/15/2026

Prepared for:

Hewlett Packard Enterprise
6280 America Center Dr
San Jose, CA 95002

Prepared By:

Gossamer Laboratories
www.gossamersec.com

1. SECURITY TARGET INTRODUCTION 3

1.1 SECURITY TARGET REFERENCE 3
1.2 TOE REFERENCE 4
1.3 TOE OVERVIEW 4
1.4 TOE DESCRIPTION 4
1.4.1 TOE Architecture 4
1.4.2 TOE Documentation 8

2. CONFORMANCE CLAIMS 9

2.1 CONFORMANCE RATIONALE 10

3. SECURITY OBJECTIVES 11

3.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 11

4. EXTENDED COMPONENTS DEFINITION 12

5. SECURITY REQUIREMENTS 13

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 13
5.1.1 Security audit (FAU) 14
5.1.2 Cryptographic support (FCS) 17
5.1.3 Identification and authentication (FIA) 21
5.1.4 Security management (FMT) 23
5.1.5 Protection of the TSF (FPT) 24
5.1.6 TOE access (FTA) 25
5.1.7 Trusted path/channels (FTP) 26
5.2 TOE SECURITY ASSURANCE REQUIREMENTS 26
5.2.1 Development (ADV) 27
5.2.2 Guidance documents (AGD) 27
5.2.3 Life-cycle support (ALC) 28
5.2.4 Tests (ATE) 29
5.2.5 Vulnerability assessment (AVA) 29

6. TOE SUMMARY SPECIFICATION 30

6.1 SECURITY AUDIT 30
6.2 CRYPTOGRAPHIC SUPPORT 31
6.3 IDENTIFICATION AND AUTHENTICATION 38
6.4 SECURITY MANAGEMENT 39
6.5 PROTECTION OF THE TSF 41
6.6 TOE ACCESS 43
6.7 TRUSTED PATH/CHANNELS 43

LIST OF TABLES

Table 1 Mobility Conductor Hardware Appliances 4
Table 2 Required Licenses 6
Table 3 Technical Decisions 9
Table 4 TOE Security Functional Components 14
Table 5 Audit Events 16
Table 6 Assurance Components 26
Table 7 Cryptographic Functions 31
Table 8 Key Exchange Methods used by TOE Services 32
Table 9: CSPs 35

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 HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0. The TOE is being evaluated as a network device.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration may be indicated by a parenthetical number placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement. Alternately, a usually descriptive textual extension may be added after a slash (/) character to identify a specific iteration. For example, iterations of a requirement such as FCS_COP.1 might be identified as FCS_COP.1/HASH and FCS_COP.1/CRYPT.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). An exception to this marking convention is the case of tables that have cells that contain equivalent to nothing (e.g., "None", "No events specified", "NA", "No additional information") where the cell is simply left blank since that represents no meaningful change but is effectively a refinement. Another exception to this marking convention is where extraneous punctuation (e.g., extra brackets) may be omitted or incorrect punctuation (e.g., extra or missing periods) may be corrected, so long as the meaning is not changed.
- 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 – HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0 Security Target

ST Version – Version 1.0

ST Date – 10/15/2026

1.2 TOE Reference

TOE Identification – The TOE is HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0 and the following required software licenses:

- Advanced Cryptography

The TOE includes the following hardware models:

HPE Networking Hardware Mobility Conductor Hardware Appliances

Product Model	CPU
MCR-HW-1K	Intel Xeon E5-2609v4 (Broadwell)
MCR-HW-5K	Intel Xeon E5-2620v4 (Broadwell)
MCR-HW-10K	Intel Xeon E5-2650v4 (Broadwell)

Table 1 Mobility Conductor Hardware Appliances

TOE Developer – Hewlett Packard Enterprise

Evaluation Sponsor – Hewlett Packard Enterprise

1.3 TOE Overview

The Target of Evaluation (TOE) is HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0. The TOE is a network device. The HPE Networking Hardware Mobility Conductor Series provides a single management layer for all HPE Networking Mobility Controller in a high scale enterprise environment.

1.4 TOE Description

The HPE Networking Hardware Mobility Conductor Series delivers the full capabilities of the HPE Networking Wireless Operating System to scale to today's enterprise needs. The Conductor enables high scale and reliability, managing up to 100,000 clients, 10,000 access points (APs) and 1,000 controllers/gateways. It also provides simplified deployment with dynamic license management, configuration hierarchy and a choice of x86 hardware appliances.

The HPE Networking Hardware Mobility Conductor Series is deployed as a conductor controller for any combination of MCR hardware appliances.

The Conductor increases scale by joining HPE Aruba Networking controllers to a Controller Cluster, improves reliability using enhanced high availability (HA), adopts configurations seamlessly based on hierarchy, and reduces or eliminates maintenance windows by enabling live upgrades.

1.4.1 TOE Architecture

HPE Networking Mobility Conductor (MCs) are hardware appliances consisting of a multicore network processor, Ethernet interfaces, and required supporting circuitry and power supplies enclosed in a metal chassis.

The HPE Aruba Networking Wireless Operating System software running on the conductors consists of two main components:

- Control Plane (CP) – implements functions which can be handled at lower speeds such as Mobility Controller system management (CLI and Web GUI), user authentication (e.g. RADIUS), Internet Key Exchange (IKE), auditing/logging (syslog) and termination of protocols operating at the system level (e.g. SSH, TLS, NTP, etc.). The Control Plane runs the Linux operating system along with various user-space applications (described below).
- Data Plane (DP) - implements functions that must be handled at high speeds such as high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging), tunnel termination (IPsec), and cryptographic

acceleration. The Data Plane runs a lightweight, proprietary real-time OS which is known as "SOS" (an acronym which used to mean "SiByte Operating System" for an earlier generation of Mobility Conductor that used the SiByte CPU). On the Conductor hardware appliances, SOS runs on separate CPU cores.

The Control Plane and Data Plane are inseparable. Administrators install the conductor software by loading a single file, identified as "HPE Aruba Networking Wireless Operating System." Internally, the conductors unpack the software image into its various components. A given HPE Aruba Networking Wireless Operating System software image has a single version number and includes all software components necessary to operate the MC appliances.

The CP runs the Linux OS, along with various custom user-space applications which provide the following CP functions:

- Monitors and manages critical system resources, including processes, memory, and flash
- Manages system configuration and licensing
- Manages an internal database used to store licenses, user authentication information, etc
- Provides network anomaly detection, hardware monitoring, mobility management, wireless management, and radio frequency management services
- Provides a Command Line Interface (CLI)
- Provides a web-based (HTTPS/TLS) management UI for the MCs
- Provides authentication services for the system management interfaces (CLI, web GUI)
- Provides IPsec key management services
- Provides network time protocol service, point to point tunneling protocol services for users, layer 2 tunneling protocol services for users, SSH services for incoming management connections, SNMP client/agent services, and protocol independent multicast (routing) services for the controller
- Provides syslog services by sending logs to the operating environment

The Linux OS running on the CP is a version 4.14.181 kernel. Linux is a soft real-time, multi-threaded operating system that supports memory protection between processes. Only OS-8.13.0 provided interfaces are used, and the CLI is a restricted command set. Administrators do not have access to the Linux command shell or operating system.

All Conductor models run the same HPE Aruba Networking Wireless Operating System 8.13.0 software and include the same cryptographic modules. Regardless of the different hardware platforms, the security functionality remains the same. The differences in the platforms are in the processing speed, throughput, memory capacity, storage, physical interfaces, number of ports, etc., and are based on performance and scalability requirements. All models run the same code with the only differences being the hardware specific code for the differently scaled hardware.

Although the TOE models have different specifications (in terms of performance and scalability), they all provide the same security functions described in the ST; therefore, they have been considered to be the same for the purposes of the ST description.

1.4.1.1 Physical Boundaries

The TOE consists of the following components:

- HPE Networking Hardware Mobility Conductor Models: MCR-HW-1K, MCR-HW-5K, MCR-HW-10K
- HPE Aruba Networking Wireless Operating System version 8.13.0

These components are identified and described in section 1.2 of this ST. The TOE is being evaluated as a physical device.

The HPE Aruba Networking Wireless Operating System (OS-8.13.0) consists of a base software package with add-on software modules that can be activated by installing the appropriate licenses. The following SFR-enforcing software modules are required to be licensed and installed in the CC evaluated configuration.

Required Software Module	Description
Advanced Cryptography	Required for SuiteB, AES-GCM and ECDSA functionality.

Table 2 Required Licenses

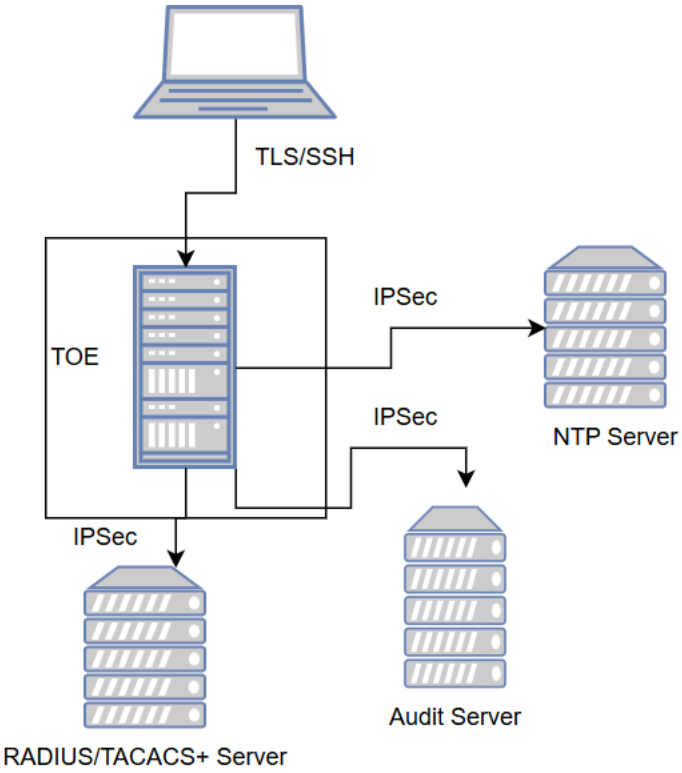


Figure 1 TOE Environmental Components

The TOE operates with the following components in the Operating Environment:

- Audit Server – The TOE utilizes an external syslog server to store audit records.
- Authentication Server – The TOE utilizes RADIUS and TACACs+ servers to authenticate users.
- Time Server – The TOE uses a Network Time Protocol (NTP) server to synchronize its system clock with a central time source.
- Web Browser – The remote administrator uses a web browser to access the Web GUI interface.
- SSH Client – The remote administrator uses an SSH client to access the CLI.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0:

- 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 including start-up and shutdown of the TOE, all administrator actions, and all events identified in Table 8 Auditable 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 in the operational environment.

1.4.1.2.2 Cryptographic support

The TOE includes cryptographic modules that provide 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 requires administrators to be identified and authenticated before they can access any TOE security functions. The TOE supports role-based authentication, so user accounts are assigned predefined roles which restrict them based on their assigned role. The TOE maintains these administrator and user attributes which can be defined locally with user names and passwords or can be defined in the context of local RADIUS or TACACS+ services. Authentication can be either locally or remotely through an external authentication server, or internally. After an administrator-specified number of failed attempts, the user account is locked out. The TOE's password mechanism provides configuration for a minimum password length. The TOE also protects, stores and allows authorized administrators to load X.509.v3 certificates for use to support authentication for IPsec connections.

1.4.1.2.4 Security management

The TOE provides the administrator role the capability to configure and manage all TOE security functions including cryptographic operations, user accounts, passwords, advisory banner, session inactivity and TOE updates. The management functions are restricted to the administrator role. The role must have the appropriate access privileges or access will be denied. The TOE's cryptographic functions ensure that only secure values are accepted for security attributes.

1.4.1.2.5 Protection of the TSF

The TOE has its own internal hardware clock that provides reliable time stamps used for auditing. The internal clock may be synchronized with a time signal obtained from an external trusted NTP server. The TOE stores passwords on flash using a SHA1 hash and does not provide any interfaces that allow passwords or keys to be read.

The TOE runs self-tests during power up and periodically during operation to ensure the correct operation of the cryptographic functions and TSF hardware. There is an option for the administrator to verify the integrity of stored TSF executable code.

The TOE includes mechanisms so that the administrator can determine the TOE version and update the TOE securely using digital signatures.

1.4.1.2.6 TOE access

The TOE allows administrators to configure a period of inactivity for administrator sessions. Once that time period has been reached while the session has no activity, the session is terminated. All users may also terminate their own sessions at any time. A warning banner is displayed at the management interfaces (Web GUI and CLI) to advise users on appropriate use and penalty for misuse of system.

1.4.1.2.7 Trusted path/channels

The TOE uses IPsec to provide an encrypted channel between itself and third-party trusted IT entities in the operating environment including external syslog server, external authentication server, and NTP server.

The TOE secures remote communication with administrators by implementing TLS/HTTPS for remote Web UI access and SSHv2 for CLI 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.

1.4.2 TOE Documentation

HPE Aruba Networking Wireless Operating System 8.13.0 Supplemental Guidance (Common Criteria Configuration Guidance), Version 3.2, October 2026

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 3.0e, 06 December 2023 (NDcPP30e)
  - Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021 (SSH10)

Package	Technical Decision	Applied	Notes
CPP_ND_V3.0E	TD1052 - NIT Technical Decision: Separation of Test Definitions for DTLSv1.2 and v1.3 (Renegotiation)	No	Requirement not claimed
CPP_ND_V3.0E	TD1033 - Sunset Dates for NDcPP Configurations	Yes	
CPP_ND_V3.0E	TD0990- NIT Technical Decision: CTR_DRBG in FCS_RBG_EXT.1.2	Yes	
CPP_ND_V3.0E	TD0973 - NIT Technical Decision: FCS_(D)TLSS_EXT.1.3 Test 2 DHE Ciphersuite Conditionality	Yes	
CPP_ND_V3.0E	TD0923 - NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Yes	
CPP_ND_V3.0E	TD0921 -. NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Yes	
CPP_ND_V3.0E	TD0900 - NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Yes	
CPP_ND_V3.0E	TD0899 - NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	Yes	
CPP_ND_V3.0E	TD0886 - Clarification to FAU_STG_EXT.1 Test 6	Yes	
CPP_ND_V3.0E	TD0880 - NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Yes	
CPP_ND_V3.0E	TD0879 - NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Yes	
CPP_ND_V3.0E	TD0868 - NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	Yes	
CPP_ND_V3.0E	TD0836 - NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Yes	
PKG_SSH_V1.0	TD1023 - Clarifications to FCS_SSH_EXT.1.5 When aes256-gcm@openssh.com Is Selected	Yes	
PKG_SSH_V1.0	TD1015 - Addition of ECD to PKG_SSH_V1.0	Yes	
PKG_SSH_V1.0	TD0967 - Allowance of Kex-strict in PKG_SSH_V1.0	Yes	
PKG_SSH_V1.0	TD0909 - Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Yes	
PKG_SSH_V1.0	TD0777 - Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
PKG_SSH_V1.0	TD0732 - FCS_SSHS_EXT.1.3 Test 2 Update	Yes	
PKG_SSH_V1.0	TD0695 - Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
PKG_SSH_V1.0	TD0682 - Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	

Table 3 Technical Decisions

2.1 Conformance Rationale

The ST conforms to the NDcPP30e/SSH10. 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 NDcPP30e/SSH10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP30e/SSH10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP30e/SSH10 should be consulted if there is interest in that material.

In general, the NDcPP30e/SSH10 has defined Security Objectives appropriate for network devices and as such are applicable to the HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0 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.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.

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

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

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

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

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

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP30e/SSH10. The NDcPP30e/SSH10 defines the following extended requirements and since they are not redefined in this ST the NDcPP30e/SSH10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP30e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP30e:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP30e:FCS_IPSEC_EXT.1: IPsec Protocol
- NDcPP30e:FCS_NTP_EXT.1: NTP Protocol
- NDcPP30e:FCS_RBG_EXT.1: Random Bit Generation
- SSH10:FCS_SSH_EXT.1: SSH Protocol
- SSH10:FCS_SSHS_EXT.1: SSH Protocol - Server
- NDcPP30e:FCS_TLSS_EXT.1: TLS Server Protocol
- NDcPP30e:FIA_PMG_EXT.1: Password Management
- NDcPP30e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP30e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP30e:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP30e:FIA_X509_EXT.3: X.509 Certificate Requests
- NDcPP30e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP30e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
- NDcPP30e:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP30e:FPT_TST_EXT.1: TSF testing
- NDcPP30e:FPT_TUD_EXT.1: Trusted update
- NDcPP30e: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 NDcPP30e/SSH10. The refinements and operations already performed in the NDcPP30e/SSH10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP30e/SSH10 and any residual operations have been completed herein. Of particular note, the NDcPP30e/SSH10 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 NDcPP30e/SSH10. The NDcPP30e/SSH10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by HPE Networking Hardware Mobility Conductor Series running HPE Aruba Networking Wireless Operating System 8.13.0 TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP30e:FAU_GEN.1: Audit Data Generation
	NDcPP30e:FAU_GEN.2: User identity association
	NDcPP30e:FAU_STG.1: Protected audit trail storage
	NDcPP30e:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP30e:FCS_CKM.1: Cryptographic Key Generation
	NDcPP30e:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP30e:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP30e:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP30e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP30e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP30e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP30e:FCS_HTTPS_EXT.1: HTTPS Protocol
	NDcPP30e:FCS_IPSEC_EXT.1: IPsec Protocol
	NDcPP30e:FCS_NTP_EXT.1: NTP Protocol
	NDcPP30e:FCS_RBG_EXT.1: Random Bit Generation
	SSH10:FCS_SSH_EXT.1: SSH Protocol
	SSH10:FCS_SSHS_EXT.1: SSH Protocol
	NDcPP30e:FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and authentication	NDcPP30e:FIA_AFL.1: Authentication Failure Management
	NDcPP30e:FIA_PMG_EXT.1: Password Management
	NDcPP30e:FIA_UAU.7: Protected Authentication Feedback
	NDcPP30e:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP30e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP30e:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP30e:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP30e:FMT_MOF.1/Functions: Management of security functions behaviour
	NDcPP30e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP30e:FMT_MOF.1/Services: Management of Security Functions Behaviour
	NDcPP30e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP30e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP30e:FMT_SMF.1: Specification of Management Functions

	NDcPP30e:FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	NDcPP30e:FPT_APW_EXT.1: Protection of Administrator Passwords
	NDcPP30e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
	NDcPP30e:FPT_STM_EXT.1: Reliable Time Stamps
	NDcPP30e:FPT_TST_EXT.1: TSF testing
	NDcPP30e:FPT_TUD_EXT.1: Trusted update
FTA: TOE access	NDcPP30e:FTA_SSL.3: TSF-initiated Termination
	NDcPP30e:FTA_SSL.4: User-initiated Termination
	NDcPP30e:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP30e:FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	NDcPP30e:FTP_ITC.1: Inter-TSF trusted channel
	NDcPP30e:FTP_TRP.1/Admin: Trusted Path

Table 4 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP30e:FAU_GEN.1)

NDcPP30e: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 Administrator 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 Administrator account shall be logged)**];*

d) Specifically defined auditable events listed in Table 5.

NDcPP30e: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.

Requirement	Audit Event	Additional Contents
NDcPP30e:FAU_GEN.1	None	None
NDcPP30e:FAU_GEN.2	None	None
NDcPP30e:FAU_STG.1	None	None.
NDcPP30e:FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
NDcPP30e:FCS_CKM.1	None	None
NDcPP30e:FCS_CKM.2	None	None
NDcPP30e:FCS_CKM.4	None	None
NDcPP30e:FCS_COP.1/ DataEncryption	None	None
NDcPP30e:FCS_COP.1/Hash	None	None

Requirement	Audit Event	Additional Contents
NDcPP30e:FCS_COP.1/ KeyedHash	None	None
NDcPP30e:FCS_COP.1/SigGen	None	None
NDcPP30e:FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
NDcPP30e:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
NDcPP30e:FCS_NTP_EXT.1	Configuration of a new time server. Removal of configured time server.	Identity if new/removed time server.
NDcPP30e:FCS_RBG_EXT.1	None	None
SSH10:FCS_SSH_EXT.1	[***Failure to establish SSH connection***] [***Establishment of SSH connection***] [***Termination of SSH connection session***] [***None***]	[***Reason for failure and non-TOE endpoint of attempted connection (IP Address)***] [***Non-TOE endpoint of connection (IP Address)***] [***Non-TOE endpoint of connection (IP Address)***] [***None***] (TD0777 applied)
SSH10:FCS_SSHS_EXT.1	None	None
NDcPP30e:FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
NDcPP30e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP30e:FIA_PMG_EXT.1	None	None
NDcPP30e:FIA_UAU.7	None	None
NDcPP30e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP30e: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.
NDcPP30e:FIA_X509_EXT.2	None	None
NDcPP30e:FIA_X509_EXT.3	None	None
NDcPP30e:FMT_MOF.1/ Functions	None	None
NDcPP30e:FMT_MOF.1/ ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP30e:FMT_MOF.1/ Services	None	None
NDcPP30e:FMT_MTD.1/ CoreData	None	None
NDcPP30e:FMT_MTD.1 /CryptoKeys	None	None
NDcPP30e:FMT_SMF.1	All management activities of TSF data.	None.
NDcPP30e:FMT_SMR.2	None	None
NDcPP30e:FPT_APW_EXT.1	None	None
NDcPP30e:FPT_SKP_EXT.1	None	None
NDcPP30e:FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).

Requirement	Audit Event	Additional Contents
NDcPP30e:FPT_TST_EXT.1	None	None
NDcPP30e:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
NDcPP30e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
NDcPP30e:FTA_SSL.4	The termination of an interactive session.	None
NDcPP30e:FTA_SSL_EXT.1	The termination of a local session by the session lock.	None
NDcPP30e:FTA_TAB.1	None	None
NDcPP30e:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	None None Reason for failure
NDcPP30e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None None Reason for failure

Table 5 Audit Events

5.1.1.2 User identity association (NDcPP30e:FAU_GEN.2)

NDcPP30e:FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Protected audit trail storage (NDcPP30e:FAU_STG.1)

NDcPP30e:FAU_STG.1.1
The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
NDcPP30e:FAU_STG.1.2
The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.1.1.4 Protected Audit Event Storage (NDcPP30e:FAU_STG_EXT.1)

NDcPP30e: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.
NDcPP30e:FAU_STG_EXT.1.2
The TSF shall be able to store generated audit data on the TOE itself. In addition [***The TOE shall consist of a single standalone component that stores audit data locally,***]
NDcPP30e:FAU_STG_EXT.1.3
The TSF shall maintain a [***log file***] of audit records in the event that an interruption of communication with the remote audit server occurs.
NDcPP30e:FAU_STG_EXT.1.4
The TSF shall be able to store [***persistent***] audit records locally with a minimum storage size of [**95,304 bytes or 349,524 bytes**].
NDcPP30e:FAU_STG_EXT.1.5
The TSF shall [***overwrite previous audit records according to the following rule: [FIFO - First in, First out]***] when the local storage space for audit data is full.
NDcPP30e:FAU_STG_EXT.1.6
The TSF shall provide the following mechanisms for administrative access to locally stored audit records [***ability to view locally***].

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP30e:FCS_CKM.1)

NDcPP30e: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] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- ECC schemes using 'NIST curves' [P-256, P-384] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;
- FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [RFC 3526]]. (TD0921 applied)

5.1.2.2 Cryptographic Key Establishment (NDcPP30e:FCS_CKM.2)

NDcPP30e:FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',
- FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [groups listed in RFC 3526]].

5.1.2.3 Cryptographic Key Destruction (NDcPP30e:FCS_CKM.4)

NDcPP30e: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 [a pseudo-random pattern using the TSF's RBG]***]

that meets the following: No Standard.

5.1.2.4 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP30e:FCS_COP.1/DataEncryption)

NDcPP30e: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, 192 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) (NDcPP30e:FCS_COP.1/Hash)

NDcPP30e:FCS_COP.1.1/Hash

The TSF shall perform cryptographic hashing services in accordance with a specified

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

5.1.2.6 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP30e:FCS_COP.1/KeyedHash)

NDcPP30e: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***] and cryptographic key sizes [**160 bit, 256 bit, 384 bit,**] and message digest sizes [***160, 256, 384***] 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) (NDcPP30e:FCS_COP.1/SigGen)

NDcPP30e: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,
- Elliptic Curve Digital Signature Algorithm] and cryptographic key sizes [
- For RSA: [modulus 2048 bits],
- For ECDSA: [256 bits, 384 bits]] that meet the following:
[***For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 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 implementing [P-256, P-384] curves that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST Recommended' curves; or FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, 'IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms', Section 6.6]. (TD0921 applied)

5.1.2.8 HTTPS Protocol (NDcPP30e:FCS_HTTPS_EXT.1)

NDcPP30e:FCS_HTTPS_EXT.1.1

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

NDcPP30e:FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

5.1.2.9 IPsec Protocol (NDcPP30e:FCS_IPSEC_EXT.1)

NDcPP30e:FCS_IPSEC_EXT.1.1

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

NDcPP30e: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.

NDcPP30e:FCS_IPSEC_EXT.1.3

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

NDcPP30e: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 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-256 (RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1***].

NDcPP30e:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [***- IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence numbers],***

and [RFC 4868 for hash functions], - IKEv2 as defined in RFC 7296 [with mandatory support for NAT traversal as specified in RFC 7296, section 2.23], and [RFC 4868 for hash functions]].

NDcPP30e: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-192, AES-CBC-256 (specified in RFC 3602)***].

NDcPP30e: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 between [1 hour] and [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 between [1 hour] and [24 hours]]]. (TD0868 applied)

NDcPP30e: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 [1 hour] and [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 [1 hour] and [8 hours]]]. (TD0868 applied)

NDcPP30e: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, 384**] bits.

NDcPP30e:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv1, IKEv2***] exchanges of length [***- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

NDcPP30e:FCS_IPSEC_EXT.1.11

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

NDcPP30e:FCS_IPSEC_EXT.1.12

The TSF shall be able to ensure 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.

NDcPP30e: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***].

NDcPP30e: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 (NDcPP30e:FCS_NTP_EXT.1)

NDcPP30e:FCS_NTP_EXT.1.1

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

NDcPP30e:FCS_NTP_EXT.1.2

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

NDcPP30e:FCS_NTP_EXT.1.3

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

NDcPP30e:FCS_NTP_EXT.1.4

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

5.1.2.11 Random Bit Generation (NDcPP30e:FCS_RBG_EXT.1)

NDcPP30e: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)***].

NDcPP30e:FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[1] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to [***the greatest security strength according to ISO/IEC 18031:2011 Table C.2 — "Security strengths, Entropy and Seed length requirements for the AES-128, 192 and 256 Block Cipher***]. (TD0990 applied)

5.1.2.12 SSH Protocol (SSH10:FCS_SSH_EXT.1)

SSH10:FCS_SSH_EXT.1.1

The TOE shall implement SSH acting as a [***server***] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [***4344, 5656, 6668, 8308, 8332***] and no other standard.

SSH10:FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods:

[***'password' (RFC 4252), 'keyboard-interactive' (RFC 4256), 'publickey' (RFC 4252): [ssh-rsa (RFC 4253)]***]

and no other methods.

SSH10:FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [***262126 bytes***] in an SSH transport connection are dropped.

SSH10:FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms:

[***aes128-ctr (RFC 4344), aes256-ctr (RFC 4344), aes128-cbc (RFC 4253), aes256-cbc (RFC 4253), aes256-gcm@openssh.com (RFC 5647)]***]

and no other mechanisms.

SSH10:FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using:

[***hmac-sha2-256 (RFC 6668)***]

and no other mechanisms.

SSH10:FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using:

[***diffie-hellman-group14-sha256, ecdh-sha2-nistp256 (RFC 5656), ecdh-sha2-nistp384 (RFC 5656)***]

and no other mechanisms.

SSH10:FCS_SSH_EXT.1.7

The TSF shall use SSH KDF as defined in [***RFC 5656 (Section 4)***] to derive the following cryptographic keys from a shared secret: session keys.

SSH10:FCS_SSH_EXT.1.8

The TSF shall ensure that [***a rekey of the session keys***] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

5.1.2.13 SSH Protocol - Server (SSH10:FCS_SSHS_EXT.1)

SSH10:FCS_SSHS_EXT.1.1
The TSF shall authenticate itself to its peer (SSH Client) using: [***rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656)***].

5.1.2.14 TLS Server Protocol (NDcPP30e:FCS_TLSS_EXT.1)

NDcPP30e:FCS_TLSS_EXT.1.1
The TSF shall implement [***TLS 1.2 (RFC 5246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:
[***/TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422,***
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422,
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/]
and no other ciphersuites.

NDcPP30e:FCS_TLSS_EXT.1.2
The TSF shall authenticate itself using X.509 certificate(s) using [***RSA with key size [2048] bits, ECDSA over NIST curves [secp256r1, secp384r1] and no other curves***].

NDcPP30e:FCS_TLSS_EXT.1.3
The TSF shall perform key exchange using: [***EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1] and no other curves***].

NDcPP30e:FCS_TLSS_EXT.1.4
The TSF shall support [***session resumption based on session tickets according to RFC 5077 (TLS 1.2)***].

NDcPP30e:FCS_TLSS_EXT.1.5
The TSF [***provides***] the ability to configure the list of supported ciphersuites as defined in NDcPP30e:FCS_TLSS_EXT.1.1.

NDcPP30e:FCS_TLSS_EXT.1.6
The TSF shall prohibit the use of the following extensions:
- Early data extension

NDcPP30e:FCS_TLSS_EXT.1.7
The TSF shall [***not use PSKs***].

NDcPP30e:FCS_TLSS_EXT.1.8
The TSF shall [***reject [TLS 1.2] renegotiation attempts***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication Failure Management (NDcPP30e:FIA_AFL.1)

NDcPP30e:FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [**1-10**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

NDcPP30e:FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing a remote session using any***

authentication method that involves a password until an Administrator defined time period has elapsed].

5.1.3.2 Password Management (NDcPP30e:FIA_PMG_EXT.1)

NDcPP30e: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 [***8***] and [***128***] characters.

5.1.3.3 Protected Authentication Feedback (NDcPP30e:FIA_UAU.7)

NDcPP30e: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 User Identification and Authentication (NDcPP30e:FIA_UIA_EXT.1)

NDcPP30e: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***].

NDcPP30e: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.

NDcPP30e:FIA_UIA_EXT.1.3

The TSF shall provide the following remote authentication mechanisms [***Web GUI password, SSH password, SSH public key***] and [***external authentication server***]. The TSF shall provide the following local authentication mechanisms [***password-based***]. (TD0900 applied)

NDcPP30e:FIA_UIA_EXT.1.4

The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in NDcPP30e:FIA_UIA_EXT.1.3.

5.1.3.5 X.509 Certificate Validation (NDcPP30e:FIA_X509_EXT.1/Rev)

NDcPP30e:FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - o Server certificates presented for DTLS/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 DTLS/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.

NDcPP30e: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.

5.1.3.6 X.509 Certificate Authentication (NDcPP30e:FIA_X509_EXT.2)

NDcPP30e:FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***IPsec***], and [***no additional uses***].

NDcPP30e:FIA_X509_EXT.2.2
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***allow the Administrator to choose whether to accept the certificate in these cases***].

5.1.3.7 X.509 Certificate Requests (NDcPP30e:FIA_X509_EXT.3)

NDcPP30e: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***].

NDcPP30e: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 (NDcPP30e:FMT_MOF.1/Functions)

NDcPP30e:FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [***modify the behaviour of***] the functions [***transmission of audit data to an external IT entity***] to Security Administrators.

5.1.4.2 Management of security functions behaviour (NDcPP30e:FMT_MOF.1/ManualUpdate)

NDcPP30e:FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

5.1.4.3 Management of Security Functions Behaviour (NDcPP30e:FMT_MOF.1/Services)

NDcPP30e:FMT_MOF.1.1/Services
The TSF shall restrict the ability to start and stop services to Security Administrators.

5.1.4.4 Management of TSF Data (NDcPP30e:FMT_MTD.1/CoreData)

NDcPP30e:FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the TSF data to Security Administrators.

5.1.4.5 Management of TSF Data (NDcPP30e:FMT_MTD.1/CryptoKeys)

NDcPP30e:FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.1.4.6 Specification of Management Functions - (NDcPP30e:FMT_SMF.1)

NDcPP30e:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [***Ability to start and stop services,***
- Ability to modify the behavior of the transmission of audit data to an external IT entity,
- Ability to manage the cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure the lifetime for IPsec SAs,
- Ability to configure the list of supported TLS ciphers,
- Ability to set the time which is used for time-stamps,
- Ability to configure NTP,
- Ability to configure the reference identifier for the peer,
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response,
- Ability to administer the TOE locally,
- Ability to configure the local session inactivity time before session termination or locking,
- Ability to configure the authentication failure parameters for FIA_AFL.1,
- Ability to manage the trusted public keys database].

(TD0880 applied)

5.1.4.7 Restrictions on Security Roles (NDcPP30e:FMT_SMR.2)

NDcPP30e:FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

NDcPP30e:FMT_SMR.2.2

The TSF shall be able to associate users with roles.

NDcPP30e:FMT_SMR.2.3

The TSF shall ensure that the conditions
- 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 (NDcPP30e:FPT_APW_EXT.1)

NDcPP30e:FPT_APW_EXT.1.1

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

NDcPP30e:FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

5.1.5.2 Protection of TSF Data (for reading of all symmetric keys) (NDcPP30e:FPT_SKP_EXT.1)

NDcPP30e: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 (NDcPP30e:FPT_STM_EXT.1)

NDcPP30e:FPT_STM_EXT.1.1

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

NDcPP30e: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 (NDcPP30e:FPT_TST_EXT.1)

NDcPP30e:FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests:
- During initial start-up (on power on) to verify the integrity of the TOE firmware and software
- Prior to providing any cryptographic service and [***at no other time***] to verify correct operation of cryptographic implementation necessary to fulfil the TSF
- [***start-up, continuous***] self-tests [**noise source health tests**]

to demonstrate the correct operation of the TSF.
(TD0836 applied)

NDcPP30e:FPT_TST_EXT.1.2

The TSF shall respond to [***all failures***] by [***rebooting***].

5.1.5.5 Trusted update (NDcPP30e:FPT_TUD_EXT.1)

NDcPP30e: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 [***the most recently installed version of the TOE firmware/software***].

NDcPP30e:FPT_TUD_EXT.1.2

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

NDcPP30e:FPT_TUD_EXT.1.3

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

5.1.6 TOE access (FTA)

5.1.6.1 TSF-initiated Termination (NDcPP30e:FTA_SSL.3)

NDcPP30e: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 (NDcPP30e:FTA_SSL.4)

NDcPP30e: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 (NDcPP30e:FTA_SSL_EXT.1)

NDcPP30e: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 (NDcPP30e:FTA_TAB.1)

NDcPP30e: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 (NDcPP30e:FTP_ITC.1)

NDcPP30e: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, [***authentication 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.

NDcPP30e:FTP_ITC.1.2

The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

NDcPP30e:FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [

- **Transmitting audit records to an audit server**
- **Sending and receiving authentication information to and from an authentication server**
- **Synchronizing time with an NTP server**

5.1.7.2 Trusted Path (NDcPP30e:FTP_TRP.1/Admin)

NDcPP30e: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.

NDcPP30e:FTP_TRP.1.2/Admin

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

NDcPP30e: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 6 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 categorisation 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 fulfil 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 all the events identified in **Table 5 Audit Events** which includes the entire packet contents for packets transmitted/received during IPsec peer session establishment. Audit records include date and time of the event, type of event, user identity that caused the event to be generated, the outcome of the event, as well as the additional content listed in column 3 of **Table 5 Audit Events**. For the administrative task of generating/import of, changing, or deleting of cryptographic keys, the key is uniquely identified in the audit records by the name assigned to it during import.

The TOE stores audit records locally and provides CLI and Web UI capabilities for the administrator to view the contents of the audit trail. For local storage, the maximum log file size for all processes is ARUBA_MAX_LOG_FILE_SIZE (i.e. 95,304 bytes). However, the security.log, system.log and user-debug logs have a maximum file size given by. A_MAX_SECURITY_USER_DEBUG_LOG_FILE_SIZE (i.e 349,524 bytes). The local protected log storage operates using the first in, first out (FIFO) method, therefore audit logs are overwritten when the available space of a specific log file is exhausted. There are 11 total logs (including the security, system, and user-debug logs), so the maximum audit storage on the TOE is 1,811,004 bytes. The local configuration is not configurable.

The TOE can also be configured to send audit records to a trusted third-party syslog server in the operational environment. The TOE uses IPsec to protect the communication channel between itself and the remote syslog server. If an external syslog server has been enabled, all audit logs are simultaneously written to both the local audit log and the syslog server. The local audit logs and logs sent to a remote server are identical.

The Security audit function satisfies the following security functional requirements:

- NDcPP30e:FAU_GEN.1: The TOE generates audit events for the not specified level of audit. Each audit record includes the date and time of the event, type of event, subject identity.
- NDcPP30e: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.
- NDcPP30e:FAU_STG.1: The TOE protects audit records in local storage from unauthorized modification or deletion. The local logs can only be viewed – they cannot be deleted or modified. There are no CLI or GUI commands for such actions
- NDcPP30e:FAU_STG_EXT.1: The TOE can be configured to export audit records to an external syslog server. The communication must be protected with an IPsec tunnel between the TOE and the syslog server. When the local storage space for audit data is full, the TOE will overwrite the previous audit records on a first in, first out (FIFO) basis

6.2 Cryptographic support

The TOE includes two cryptographic modules that provide supporting cryptographic functions. The ArubaOS Crypto Module version 1.0 provides all of the cryptographic functions implemented for IKEv2/IPsec, while the Hewlett Packard Enterprise OpenSSL 3 Provider Module version 3.1.4a provides all other cryptographic functions implemented in the TOE including those for IKEv1/IPsec, TLS and SSH.

The evaluated configuration requires that the TOE be configured in FIPS mode to ensure that the CAVP tested algorithms are used. The following functions have been CAVP certified:

Requirements	Functions	Standards	Cert
	Cryptographic key generation		
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit	FIPS Pub 186-4	A4803 A2689
FCS_CKM.1	ECC schemes using 'NIST curves' P-256, P-384	FIPS Pub 186-4	A4803 A2689
FCS_CKM.1	FFC Schemes using 'safe-prime' groups DH-14	NIST SP 800-56A	Tested with a known good implementation
	Cryptographic key establishment/distribution		
FCS_CKM.2	Elliptic curve-based key establishment schemes (P-256, P-384)	NIST SP 800-56A	A4803 A2689
FCS_CKM.2	FFC Schemes using 'safe-prime' groups DH-14	NIST SP 800-56A	Tested with a known good implementation
	Encryption/Decryption		
FCS_COP.1/ DataEncryption	AES CBC (128, 192 and 256 bits)	FIPS Pub 197 NIST SP 800-38A	A4803 A2689
FCS_COP.1/ DataEncryption	AES GCM (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38D	A4803 A2689
FCS_COP.1/ DataEncryption	AES CTR (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38A	A4803
	Cryptographic signature services		
FCS_COP.1/SigGen	RSA Digital Signature Algorithm (modulus 2048)	FIPS Pub 186-4	A4803 A2689
FCS_COP.1/SigGen	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve size of 256 or 384 bits	FIPS Pub 186-4	A4803 A2689
	Cryptographic hashing		
FCS_COP.1/Hash	SHA-1/256/384/512 (digest sizes 160, 256, 384 and 512 bits)	FIPS Pub 180-3	A4803 A2689
	Keyed-hash message authentication		
FCS_COP.1/ KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 (key and output MAC sizes 160, 256, and 384, respectively))	FIPS Pub 198-1 FIPS Pub 180-3	A4803 A2689
	Random bit generation		
FCS_RBG_EXT.1	CTR_DRBG (AES) with SW based noise sources (256 bits)	NIST SP 800-90	A4803

Table 7 Cryptographic Functions

The TOE fulfills all of the FIPS PUB 186-4 requirements for cryptographic key generation without extensions. The TOE conforms to all shall, shall-not, should and should-not statements. For ECDSA, the TOE implements section B.4.2 in Appendix B of FIPS PUB 186-4. The TOE implementation of Diffie-Hellman group 14 (2048 MODP) meets RFC 3526, Section 3.

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

Table 8 Key Exchange Methods used by TOE Services

The TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. Zeroization is accomplished by overwriting the secret or private key with all zeroes. The table below identifies all secret and private keys and Critical Security Parameters (CSPs), the related zeroization procedures and whether any interface is available to view the plaintext key.

CSP	CSPs type	Generation and Use	Storage	Zeroization
Key Encryption Key (KEK)	Triple-DES (192 bits)	Hardcoded during manufacturing. Used only to protect keys stored in the flash, not for key transport.	Stored in Flash memory (plaintext).	Zeroized by using command 'zeorize-tpm-keys' and 'wipe out flash'
Diffie-Hellman private key	Diffie-Hellman Group 14 (224 bits)	Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
Diffie-Hellman public key	Diffie-Hellman Group 14 (2048 bits)	Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
Diffie-Hellman shared secret	Diffie-Hellman Group 14 (2048 bits)	Established during Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
EC Diffie-Hellman private key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
EC Diffie-Hellman public key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
EC Diffie-Hellman shared secret	EC Diffie-Hellman (Curves: P-256 or P-384)	Established during EC Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device

CSP	CSPs type	Generation and Use	Storage	Zeroization
RADIUS server shared secret	8-128 characters shared secret	Entered by CO role. Used for RADIUS server authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
Enable secret	8-32 characters password	Entered by CO role. Used for CO role authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
User Password	8-32characters password	Entered by CO role. Used for User role authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
IKEv1 Pre-shared secret	Shared secret (8 - 64 ASCII or 64 HEX characters)	Entered by CO role. Used for IKEv1 peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
skeyid	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was established via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving other keys in IKE protocol implementation.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device.
skeyid_d	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
SKEYSEED	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv2) and it will be used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
IKE session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	The IKE session (IKE Phase I) authentication key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKEv1/IKEv2 payload integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device

CSP	CSPs type	Generation and Use	Storage	Zeroization
IKE session encryption key	Triple-DES (192 bits, 3 Key, CBC) /AES (128/192/256 bits, CBC)	The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKE payload protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
IPSec session encryption key	Triple-DES (192 bits, 3 Key, CBC) / AES and AES-GCM (128/256 bits, CBC) NOTE: 192 bit CAVP tested, but not used.	The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used to secure IPsec traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
IPSec session authentication key	HMAC-SHA-1 (160 bits)	The IPsec (IKE Phase II) authentication key. This key is derived via using the KDF defined in SP800-135 KDF (IKEv1/IKEv2). Used to verify the integrity of IPsec traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
SSHv2 session key	AES (128/192/256 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to secure SSHv2 traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
SSHv2 session authentication key	HMAC-SHA-1 (160-bit)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to verify the integrity of SSHv2 traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
TLS pre-master secret	48 bytes secret	This key is transferred into the module, protected by TLS RSA public key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
TLS session encryption key	AES 128/192/256 bits	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to secure TLS traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
TLS session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to verify the integrity of TLS traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
RSA Private Key	RSA 2048 bit private key	This key is generated in the module. Used for IKEv1, IKEv2, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command ‘write erase all’

CSP	CSPs type	Generation and Use	Storage	Zeroization
RSA public key	RSA 2048 bits public key	This key is generated in the module. This Key can also be entered by the CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv1, IKEv2, TLS, OCSP (verifying OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all'
ECDSA Private Key	ECDSA suite B P-256 and P-384 curves	This key is generated in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all'
ECDSA Public Key	ECDSA suite B P-256 and P-384 curves	This key is generated in the module. This Key can also be entered by the CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all'.
Factory CA Public Key	RSA (2048 bits)	This is RSA public key. Loaded into the module during manufacturing. Used for Firmware verification.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all'

Table 9: CSPs

The supporting cryptographic functions are included to support the HTTPS/TLS (RFCs 2818, TLS 1.2 (RFC 5246), SSHv2 (RFCs 4251, 4252, 4253, 4254 and 4344), and IPsec (RFC 4301) secure communication protocol. In the FIPS mode of operation, the cipher parameters have been hardcoded to use only the Common Criteria evaluated configuration and are not configurable by the administrator. No optional protocol characteristics are implemented.

The TOE supports TLSv1.2. Any other SSL/TLS versions are not supported by the TOE and such connection attempts will be rejected. Remote administration via the Web UI is protected using TLS/HTTPS. The following cipher suites are implemented by the TOE, and are configurable:

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422,

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422,

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_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422,

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422,

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

The TOE performs ECDSA key establishment with curves secp256r1, and secp384r1. The TOE is capable of authenticating itself to clients with RSA certificates with key size 2048 as well as ECDSA certificates using NIST curve secp256r1 and secp384r1. For its HTTPS/TLS server, the TOE supports session resumption using session tickets that meet the format identified in section 4 of RFC 5077. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm. If a client hello contains a session ID based on a session ticket issued previously by the server, the server will check to see if the context of the session has not expired. If the context of the session is still available and the TOE accepts the resumption attempt, then the security context of the new connection is tied to the original connection and a full handshake is not required.

Remote administration via the Command Line Interface (CLI) is protected using SSHv2. The TOE supports SSHv2 with AES CTR 128 or 256 bit ciphers, AES CBC 128 or 256 bit ciphers, and AES GCM 256 bit cipher, in conjunction with HMAC-SHA2-256 and RSA using the following key exchange methods: diffie-hellman-group14-sha256, ecdh-sha2-nistp256, and ecdh-sha2-nistp384. The TOE also supports rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256 for server authentication. While other ciphers and hashes are implemented in the product, they are disabled while the TOE is operating in FIPS mode.

SSHv2 supports, public-key, password-based, and keyboard-interactive authentication and can be configured. Keyboard-interactive, while supported only allows authentication with a password, and does not support multifactor authentication. SSHv2 connections are rekeyed after a period of no longer than one hour or no more than 512 megabytes of transmitted data or 512 megabytes of received data. All of these thresholds are checked by the TOE and rekeying is performed upon reaching the threshold that is hit first. Password based authentication can be configured. The authentication timeout period is 30 seconds allowing clients to retry only 3 times. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. The TOE limits packets to 262126 bytes. 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 (262126 bytes) the packet will be dropped.

The TOE includes an implementation of IPsec in accordance with RFC 4301 for security. The TOE supports IPsec in transport mode and tunnel mode. The IPsec ESP protocol is implemented in conjunction with AES-CBC-128, AES-CBC-192 and AES-CBC-256 (as specified by RFC 3602) and AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106) together with the HMAC-SHA-1 algorithm.

The TOE implements both IKEv1, as defined in RFCs 2407, 2408, 2409, and RFC 4109; and IKE2, with support for NAT traversal, as defined in RFC ***7296***and RFC 4868 for hash functions (HMAC-SHA-1). 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 uses the AES-CBC-128, AES-CBC-192 and AES-CBC-256 algorithms as specified in RFC 3602 to encrypt the IKEv1 or IKEv2 payload. Note that aggressive mode is not used with IKEv1, only main mode is supported.

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). The TOE supports the following PRF hash functions: PRF_HMAC_SHA1, PRF_HMAC_SHA256 and PRF_HMAC_SHA384 and generates nonces used in the IKEv1 and IKEv2 exchanges of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash. Nonces are generated using the function RANDOM_numberGenerator() which generates numbers that meet the requirements specified in FCS_RBG_EXT.1 for random bit generation.

The TOE supports SA lifetime limits based on length of time for both IKEv1 and IKEv2. Lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) and IKEv2 SAs are established during administrator configuration of the IKE policies. In the case of IKEv1, lifetimes can be configured based on length of time within 1-24 hours for phase 1 and within 1-8 hours for phase 2 by specifying the number of seconds for the SA lifetime. In the case of IKEv2, lifetimes for the IKEv2 IKE_SA can be configured by specifying the number of seconds within 1-24 hours. For the IKEv2 IKE_CHILD SA, lifetimes can be configured by specifying the number of seconds within 1-8hrs for the SA lifetime.

In the IKEv1 phase 1 and phase 2 and IKEv2 IKE_SA and IKE_CHILD exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE initiates the IKE negotiation, the DH group is sent in order

according to the peer's configuration. When the TOE receives an IKE proposal, it will select the first match and the negotiation will fail if there is no match.

The Administrator is responsible for ensuring that IKE/IPsec policies are configured so that the strength of the negotiated symmetric algorithm (in terms of the number of bits in the key) in the IKEv1 Phase 2 SA or IKEv2 CHILD_SA is less than or equal to the strength of the IKEv1 Phase 1 SA or IKEv2 IKE_SA. Administrators should configure IKE/IPsec policies so that the strength of the IKE association is greater than or equal to the strength of the IPsec tunnel (for example, by always using AES-256). However, if a misconfiguration is made, the TOE will reject the security association along with generating an audit log message.

Pre-shared keys used for IPsec can be constructed of essentially any alphabetic character (upper and lower case), numerals, and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 6-160 characters in length (e.g., 22 characters). The TOE requires suitable keys to be entered by an authorized administrator using a Web GUI or CLI function.

The TOE will only establish a trusted IPsec channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following type: Distinguished Name (DN). Fields within the DN are not individually selectable; the DN must be an exact match for the entire DN string.

The TOE implements an SPD and processes packets to satisfy the behavior of DISCARD, BYPASS, and PROTECT packet processing as described in RFC 4301 to determine what traffic gets protected with IPsec, what gets bypassed, and what gets dropped. Each packet is either PROTECTed using IPsec security services, DISCARDed, or allowed to BYPASS IPsec protection, based on the applicable SPD policies. The SPD is achieved via the routing table and firewall policies. The TOE administrator implicitly configures the IPsec SPD via the routing table and firewall policies. The TOE compares packets against the configured rules to determine if any of the packets match the rules. The packets can be matched based upon source IP address, destination IP address, protocol type (e.g., TCP, UDP, ICMP). Traffic not matching any rule is passed to the next stage of processing. The TOE includes a final rule that causes the network packet to be discarded if no other rules are matched.

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP30e:FCS_CKM.1: The TOE supports cryptographic key generation for the following schemes: RSA schemes using key sizes of 2048 bits, ECC schemes using NIST curves P-256 and P-384, and FFC schemes using Diffie-Hellman group 14 cryptographic key derivation.
- NDcPP30e:FCS_CKM.2: The TOE supports cryptographic key establishment for Elliptic curve based key establishment schemes and key establishment schemes using Diffie-Hellman group 14.
- NDcPP30ee:FCS_CKM.4: Keys are zeroized when they are no longer needed by the TOE.
- NDcPP30e:FCS_COP.1/DataEncryption: The TOE supports AES CBC (128, 192 and 256 bits), AES CTR (128 and 256 bits), and AES GCM (128 and 256 bits) for data encryption/decryption.
- NDcPP30e:FCS_COP.1/Hash: The TOE supports SHA-1/256/384/512 (digest sizes 160, 256, 384, and 512 bits) for cryptographic hashing. The TOE implements SHA-1, SHA-256 and SHA-384 in support of TLS v1.2 in conjunction with AES (CBC and GCM) 128- or 256-bit ciphers and RSA and ECDSA and SHA-256 and SHA-512 in support of SSH.
- NDcPP30e:FCS_COP.1/KeyedHash: The TOE supports HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 (key and output MAC sizes 160, 256, 384, respectively) for keyed-hash message authentication. The TOE implements HMAC-SHA-1and diffie-hellman-group 14-sha1 in support of SSHv2 and HMAC-SHA-1 in support of IPsec. HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 are all used in support of TLS.
- NDcPP30e:FCS_COP.1/SigGen: The TOE supports RSA (modulus 2048) and ECDSA with elliptical curve size 256 or 384 bits for signature generation and verification.

- NDcPP30e:FCS_HTTPS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions in compliance with RFC 2818. The TOE ensures that a TLS negotiation is performed prior to any HTTP communication which is encrypted over the TLS channel.
- NDcPP30e:FCS_IPSEC_EXT.1: The TOE supports IPsec cryptographic network communication protection (RFC 4868, RFC 4945).
- NDcPP30e:FCS_NTP_EXT.1: The TOE updates its system time using IPsec to provide a trusted communication path between itself and an NTP v4 server.
- NDcPP30e:FCS_RBG_EXT.1: The TOE instantiates its AES-256 CTR_DRBG with a 512-bit seed (containing a minimum of 256 bits of entropy) from one software-based noise source.
- SSH10:FCS_SSH_EXT.1 FCS_SSHS_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- NDcPP30e:FCS_TLSS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions

6.3 Identification and authentication

The TOE supports role-based authentication. Users can authenticate to an external authentication server or to the Conductor's internal database.

The administrator can create a user account in the internal database and assign a predefined role to that account. Users logging in to the Conductor are restricted based on their assigned role. In this case, the authentication mechanism is provided by the TOE and the credentials are maintained in the internal database. The administrator can also configure the TOE so that users are authenticated using an external authentication server. The TOE supports RADIUS and TACACS+ servers. A trusted channel using IPsec is established between the TOE and an external authentication server.

Remote administrators are configured as users who have privileges to access the CLI and Web GUI administration interfaces and who are authenticated as users using the local database or an external authentication server. The remote administrators authenticate as users using a username and password via Web GUI and username/password or public key for SSH. The Web GUI interface provides a trusted path to connect to the TOE via HTTPS. The HTTPS interface uses a server RSA or ECDSA certificate which is stored on the TOE. SSH provides a trusted path to connect to the CLI. It uses SSH_RSA for public keys. Direct console to the CLI only supports username/password. A successful logon takes place when a recognized username/password combination is provided.

The Conductor maintains a counter of failed authentication attempts for a given administrative username within the past three minutes. An unsuccessful authentication attempt is detected when an invalid password or public key is entered for a valid username. If the failed authentication threshold is reached for a given username, that user account is locked out until the configured lock-out period has expired. The failed authentication threshold is enforced by the TOE and when using an external authentication server. In order to ensure that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, the serial port does not enforce account locking. Therefore, local administrators accessing the TOE via the local console will always have access.

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

- NDcPP30e:FIA_AFL.1: After an administrator specified (1-10) number of failed attempts, the TOE will lockout (denylist) the offending remote administrator and log the event. The offending administrator will remain locked out until the administrator configured lock-out period has expired. FIA_AFL.1 is enforced by the TOE and when using an external authentication server.
- NDcPP30e:FIA_PMG_EXT.1: The TOE authentication mechanism provides configuration for minimum password length. The following calculation is based on the following facts:
  - Password is case-sensitive
  - A-Z, a-z, 0-9, !@#$%^&*()_+, and extended characters
  - Password minimum length is set to 8 and the maximum length is 128

- Passwords have maximum lifetime and new passwords must contain a minimum of 4 character changes from the previous password

Passwords must be at least eight characters long. Numeric, alphabetic (upper and lower case), and keyboard/extended characters can be used, which gives a total of 95 characters to choose from. An eight character password using all characters has 95^8 total possible combinations. The probability for a random attempt to succeed is therefore less than one in 1,000,000,000,000,000

- NDcPP30e:FIA_UAU.7: The TOE provides only obscured feedback to the administrative user while authentication is in progress at the local console by displaying an asterisk (*) for each character entered.
- NDcPP30e:FIA_UIA_EXT.1: The TOE provides local accounts and can also be configured to utilize a RADIUS or TACACS+ authentication server in its operational environment. The administrator can configure the TOE to provide the same or different authentication mechanism (local, remote) for wireless users and administrators. The TOE shall invoke the correct authentication mechanism as configured by the administrator. Prior to requiring the non-TOE entity to initiate the identification and authentication process, the TOE displays an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE (FTA_TAB.1). The TOE requires an administrator to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user.
- NDcPP30e:FIA_X509_EXT.1/Rev: The TOE protects, stores and allows authorized administrators to load X.509v3 certificates for use to support authentication for IPsec connections. Certificates are loaded into the Conductor using the System -> Certificates panel of the Web-based user interface. The Conductor supports loading of certificates in PEM, DER, or PFX format. Private keys may be loaded onto the Conductor through a password-protected PFX file or may originate on the Conductor at the time a Certificate Signing Request (CSR) is generated.

  During runtime, certificates and private keys are stored in ramdisk, in volatile memory, in decrypted form. This allows private keys to be accessed rapidly for high network load conditions. When powered off, private keys are stored encrypted in non-volatile (flash) memory. The encryption method used is AES256 as described in the CSP table.

  Certificates are validated as part of the authentication process when they are presented to the TOE, when performing trusted updates and when they are loaded into the TOE. The TOE validates certificates in accordance with RFC 5280 certificate validation rules. The TOE validates the revocation status of certificates using OCSP as specified in RFC 6960.

  OCSP checking is supported. OCSP responder servers must be configured in the revocation profile. The profile consists of the server URL, the responder cert (used to verify OCSP responses), and the action to take if the OCSP responder is non-responsive (permit or deny). If revocation checking is enabled, all certs in the chain except for the root are verified in order, starting with the peer cert and ending at the penultimate CA certificate. Certificate validation includes verification of the basicConstraints extension and the CA flag to determine whether they are present and set to TRUE. The check also ensure that the key usage extension is present. OCSP certificates must have the OCSP signing purpose in the extendedKeyUsagefield.
- NDcPP30eFIA_X509_EXT.2: The TOE uses X509v3 certificates to support authentication for IPsec. If the TOE cannot determine the validity of a certificate, the administrator has the option of choosing whether or not to accept the certificate.
- NDcPP30e:FIA_X509_EXT.3: The TOE generates Certificate Request Messages and includes the following information: public key, common name, organization, organizational unit, country. Upon receiving the CA Certificate response, the TOE will validate the chain of certificates from the Root CA.

6.4 Security management

The TOE provides the administrator role the capability to enable the management of TOE security attributes, TSF data and TOE security functions. The administrator can configure TOE security settings and policies using the Web UI via HTTPS, or the command line interface via serial console locally or remotely using SSHv2. The Web GUI is a just a front-end to the CLI (i.e., calls the CLI internally). It provides a user-friendly interface for the administrator to

manage the TOE. Every function that can be performed on the Web GUI, can also be performed using the CLI but not vice versa. However, every security management function claimed can be done either using the Web GUI or CLI.

The TOE supports role-based authentication. There are three types of roles: administrator[1] role, limited administrator role, and wireless user role. The limited administrator role and wireless user role are not TOE Security management roles. The administrator can manage the TOE using the Web GUI or CLI. Wireless clients cannot access the TOE through the Web GUI or CLI interfaces and, therefore, do not have access to the management functionalities of the TOE. The limited administrator role can perform non-security tasks only.

The administrator/remote administrator authenticates with a username and password via an HTTPS connection or via the interactive command line. Once the administrator is authenticated, the Conductor provides management interfaces which can be used by the administrator to configure the TOE security functions. Local administrators can also use the CLI via a serial console (direct) connection to the TOE by using username and password. Remote administrators may use the Web GUI interface from the browser or may also use the CLI interface via an SSH protocol connection from an SSH client. Administrators have the same role-based privileges regardless of connecting locally or remotely.

The TOE provides the administrator with capabilities to manage all security functions identified in this Security Target, including the following:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to start and stop services,
- Ability to modify the behavior of the transmission of audit data to an external IT entity,
- Ability to manage the cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure the lifetime for IPsec SAs,
- Ability to set the time which is used for time-stamps,
- Ability to configure NTP,
- Ability to configure the reference identifier for the peer,
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response,
- Ability to administer the TOE locally,
- Ability to configure the local session inactivity time before session termination or locking,
- Ability to configure the authentication failure parameters for FIA_AFL.1,
- Ability to manage the trusted public keys database,

The Security management function satisfies the following security functional requirements:

- NDcPP30e:FMT_MOF.1/Functions: Only Security Administrators can modify the behavior of the transmission of audit data to an external audit server.
- NDcPP30e:FMT_MOF.1/ManualUpdate: Only Security Administrators can enable the function to perform a manual update.
- NDcPP30e:FMT_MOF.1/Services: Only Security Administrators can enable and disable TOE functions and services. An administrator can start and stop remote syslog services, certificate validation checks, IPsec, TLS, SSH, NTP polling, authentication services, and certificate validation. These services can be enabled or disabled through policy configuration.
- NDcPP30e:FMT_MTD.1/CoreData: Only Security Administrators can manage TSF data. There are no security functions available through any interfaces prior to administrator login. Non-administrative users (ie. wireless users) do not have access to the TOE via the Web UI or the CLI, therefore, they do not have any access to the security functions of the TOE.

[1] Some HPE documents may refer to as the “root” and/or “crypto officer” role. This role fulfills the role of ‘Security Administrator’ as identified in FMT_SMR.2

- NDcPP30e:FMT_MTD.1/CryptoKeys: Only Security Administrators can manage the cryptographic keys and certificates used for IPsec operations.
- NDcPP30e:FMT_SMF.1: The TOE provides administrative functions to manage all TOE security functions identified in this Security Target.
- NDcPP30e:FMT_SMR.2: The TOE maintains the security role of Security Administrator who can manage the TOE both remotely and locally. The TOE supports role-based authentication. Administrators can make use of both local and remotely accessible administrator interfaces.

6.5 Protection of the TSF

The TOE has an internal hardware clock that provides reliable time stamps used for auditing. The internal clock may be synchronized with a time signal obtained from an external NTP server. Note that the clock is used primarily to provide a timestamp for audit records, but is also used to support timing elements of cryptographic functions, certificate validity checks, session timeouts, and unlocking of administrator accounts locked as a result of authentication failure.

The TOE runs a suite of self-tests during power-up and periodically during operation, which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. An administrator can choose to reboot the TOE to perform power-up self-tests. The Conductor runs the suite of FIPS 140-3 validated cryptographic module self-tests during start-up or on request from the administrator, including immediately after generation of a key (FIPS self-tests, including the continuous RNG test).

The following tests are performed:

- Hewlett Packard Enterprise OpenSSL 3 Provider Module:
  - AES Known Answer Tests (KAT)
  - Triple-DES KAT
  - RNG KAT
  - RSA KAT
  - ECDSA (sign/verify)
  - SHA (SHA1, SHA256 and SHA384) KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256 and HMAC-SHA384) KAT
- HPE ArubaOS Crypto Module
  - AES KAT
  - Triple-DES KAT
  - SHA (SHA1, SHA256, SHA384 and SHA512) KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC- SHA512) KAT
  - RSA (sign/verify)
  - ECDSA (sign/verify)
  - FIPS 186-2 RNG KAT
- HPE Aruba Networking Uboot BootLoader Module
  - Firmware Integrity Test: RSA 2048-bit Signature Validation
- HPE Aruba Networking Hardware Known Answer Tests:
  - AES KAT
  - AES-CCM KAT
  - AES-GCM KAT
  - Triple DES KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT

The following Conditional Self-tests are performed by the TOE:

- **Continuous Random Number Generator Test.** This test is run upon generation of random data by the switch's random number generators to detect failure to a constant value. The module stores the first random number for subsequent comparison, and the module compares the value of the new random number with the random number generated in the previous round and enters an error state if the comparison is successful.
- **Bypass test.** Ensures that the system has not been placed into a mode of operation where cryptographic operations have been bypassed, without the explicit configuration of the cryptographic officer. To conduct the test, a SHA1 hash of the configuration file is calculated and compared to the last known good hash of the configuration file. If the hashes match, the test is passed. Otherwise, the test fails (indicating possible tampering with the configuration file) and the system is halted.
- **RSA Pairwise Consistency test.** When the TOE generates a public and private key pair, it carries out pair-wise consistency tests for both encryption and digital signing. The test involves encrypting a randomly-generated message with the public key. If the output is equal to the input message, the test fails. The encrypted message is then decrypted using the private key and if the output is not equal to the original message, the test fails. The same random message is then signed using the private key and then verified with the public key. If the verification fails, the test fails.
- **ECDSA Pairwise Consistency test.** See above RSA pairwise consistency test description.
- **Firmware Load Test.** This test is identical to the Uboot BootLoader Module Firmware Integrity Test, except that it is performed at the time a new software image is loaded onto the system. Instead of being performed by the BootLoader, the test is performed by the HPE Aruba Networking Wireless operating system. If the test fails, the newly loaded software image will not be copied into the image partition, and instead will be deleted.
- **Known-answer tests (KAT)** involve operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test shall fail.

If a self-test fails, the TOE will immediately halt operation and enter an error state thereby preventing potentially insecure operations (i.e., maintaining a secure state). The Conductor will reboot after a self-test failure. During reboot, memory is re-initialized, which wipes all keys and user data. If a self-test failure continues to occur, the Conductor will continue to reboot repeatedly and will require return to manufacturer.

The above tests are sufficient to demonstrate that the TSF is operating correctly by verifying the integrity of the TSF and the correct operation of cryptographic components.

The TOE uses IPsec, TLS, and SSH for all communications terminating at the TOE. Each of these protocols includes features to detect replayed data and the TOE will reject any such data that is received.

In order to update the firmware, the administrator can download the firmware file from the HPE support website. Using TFTP, FTP, SCP, or HTTPS (Web UI only), the administrator can import the image into the Conductor.

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

- NDcPP30e:FPT_APW_EXT.1: Passwords are not stored in plaintext. They are stored in flash using a SHA1 hash. Note that for OS-8.13.0, flash is an SCSI or IDE disk partition. In all cases, passwords cannot be viewed through an interface designed specifically for that purpose.
- NDcPP30e:FPT_SKP_EXT.1: The TOE provides no interfaces that allow pre-shared, symmetric or private keys to be read. Section 6.2 describes how the pre-shared keys, symmetric keys and private keys are stored.
- NDcPP30e:FPT_STM_EXT.1: The TOE relies on external time and date information either provided manually by the Security Administrator or through the use of an external trusted NTP server.
- NDcPP30e:FPT_TST_EXT.1: The TOE offers a suite of self-tests to verify the correct operation of the key generation and static TSF cryptographic data. Firmware integrity tests verify the digital signature of the code image using RSA 2048 bit signature validation. Software images are cryptographically signed, and an image with an invalid signature will not be copied by the controller into the image partition. Similarly, a software

image stored in the image partition through external means will be rejected by the hardware bootloader if the image signature is invalid.

- NDcPP30e:FPT_TUD_EXT.1: The Conductor allows administrators to query the current version of its firmware/software using the 'show image version' command and allows those administrators to manually initiate firmware/software updates. Prior to installing any update, the administrator can verify the digital signature of the update. Administrators can update the TOE executable code using image files manually downloaded from the HPE Networking support portal. The administrator may perform an update from either the WebUI or CLI. Upgrade instructions are documented in the release notes for each software release, which will be posted in the same directory as the image file on the support portal.

  The Controller contains two partitions which store system images. Loading an image into one of these partitions constitutes installation with delayed activation. At this point, the image is not yet active. The loaded image becomes active after the administrator specifies the partition to be used for boot, and the TOE is rebooted. Upon booting, the TOE loads the image in the selected partition, and the inactive version becomes active.

  The update image is digitally signed using RSA 2048-bit signature validation. When an update is initiated, the TOE verifies the digital signature with the stored public root CA certificate which is programmed into the boot ROM of all HPE Networking products at the time of manufacturing. Upon successful verification, the TOE boots using the new image. Should verification fail, the TOE will enter into an error state. The TOE's error state will allow direct console access only, where an administrator can change to a new file partition or TFTP a new image and re-boot.

6.6 TOE access

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

The TOE terminates remote or local administrator sessions after session inactivity time exceeds a configurable session idle timeout. The session idle timeout is the maximum amount of time an administrator may remain idle.

All users may also terminate their own sessions at any time simply by logging off their session.

The TOE access function satisfies the following security functional requirements:

- NDcPP30e:FTA_SSL.3: By default, the TOE will terminate remote interactive sessions after a configurable time interval of session inactivity.
- NDcPP30e:FTA_SSL.4: Administrative users can log off at any time by issuing the applicable command.
- NDcPP30e:FTA_SSL_EXT.1: Local inactive administrator sessions on the TOE are terminated, just like remote inactive administrator sessions, after the configured timeout period.
- NDcPP30e:FTA_TAB.1: The TOE displays an advisory warning banner regarding use of the TOE prior to establishing an administrator session. The administrator can configure the warning message displayed in the banner.

6.7 Trusted path/channels

The TOE provides trusted paths for remote administration and trusted channels for communication between itself and peers in the operating environment.

For remote administrators, the TOE uses HTTPS/TLS to offer secure remote web GUI-based administration and SSH to offer a secure remote administration CLI.

The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and the external authentication server, syslog server, and NTP server. To configure the channels, the administrator uses the Configuration -> Services -> VPN panel of the Web GUI to create the host-to-host IPsec/IKE connections. All configuration settings must specify CAVP tested encryption algorithms as specified by the FCP_COP.1 requirements.

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

- NDcPP30e:FTP_ITC.1: The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and an external authentication server, syslog server, and NTP server.
- NDcPP30e:FTP_TRP.1/Admin: The TOE uses SSH, TLS/HTTPS to provide a trusted communication path between itself and 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 detection of modification of the communicated data.