Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6465T, 6560, 6570, 6860N, 6865, 6870, 6900, 9900 with AOS 8.10 Security Target

Version:	1.2
Status:	Final
Last Update:	2026-01-15
Validation ID	11627
Classification:	Public
Prepared for:	ALE USA Inc.
Prepared by:	atsec information security corporation

Legal Notice

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2025-08-11	atsec	First version.
1.1	2025-09-22	atsec	Improved as per ECR comments.
1.2	2026-01-15	atsec	Added TD0967.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6465T, 6560, 6570, 6860N, 6865, 6870, 6900, 9900 with AOS 8.10 Security Target
Version:	1.2
Status:	Final
Date:	2026-01-15
Sponsor:	ALE USA Inc.
Developer:	ALE USA Inc.
Validation Body:	NIAP-CCEVS
Validation ID:	11627
Keywords:	ALE, Switch, OmniSwitch, AOS

1.2 TOE Identification

The Target of Evaluation (TOE) is Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6465T, 6560, 6570, 6860N, 6865, 6870, 6900, 9900 with AOS 8.10.

1.3 TOE Type

The TOE type is Network Device.

1.4 TOE Overview

The TOE is Alcatel-Lucent Enterprise OmniSwitch family of network switches. All the switches in the OmniSwitch family use the same secure level code, support Deep Packet Inspection (DPI) and Shortest Path Bridging (SPB), and provide wire-rate switching and routing capacity.

The TOE includes hardware and software components. The following hardware platforms are covered in this evaluation:

- OmniSwitch 6360 (OS6360)
- OmniSwitch 6465 (OS6465)
- OmniSwitch 6465T (OS6465T)
- OmniSwitch 6560 (OS6560)
- OmniSwitch 6570 (OS6570)
- OmniSwitch 6860N (OS6860N)
- OmniSwitch 6865 (OS6865)
- OmniSwitch 6870 (OS6870)
- OmniSwitch 6900 (OS6900)
- OmniSwitch 9900 (OS9900)

The software running all the platforms above is Alcatel-Lucent Operating System (AOS), which is the single purpose operating system supporting device management and network management. The build number of the TOE software tested in this evaluation is AOS 8.10.13.R12.

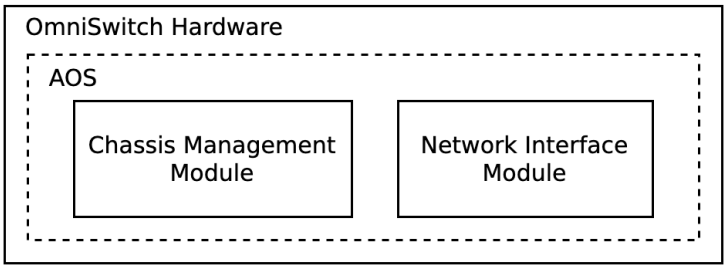
Outlined below are the major security features of the TOE:

- Security Audit. The TOE generates audit data which is stored locally and remotely.
- Cryptographic Support. The TOE implements standard cryptographic algorithms underlying the other security features.
- Identification and Authentication. The TOE authenticates the administrators accessing the TOE through the local console or over the network.
- Security Management. The TOE provides a Command-Line Interface (CLI) to the administrators for managing the configurable aspects of the TOE.
- Protection of the TSF. The TOE protects cryptographic keys and passwords, verifies the TOE software updates, performs self-tests, and maintains reliable time stamps.
- TOE Access. For both local and remote administrative sessions, the TOE presents an advisory notice upon session initiation and terminates idle sessions.
- Trusted Path/Channels. The TOE implements standard cryptographic protocols to secure the communications with other trusted IT entities as well as the administrative sessions over the network.

1.5 TOE Description

The following diagram shows the major components of the TOE.

Figure 1: TOE Architecture



Chassis Management Module (CMM) refers to the functionalities of control plane, including administrative interfaces, cryptographic functions, configuration management, and software updates, etc. Network Interface Module (NIM) refers to the functionalities of data plane, providing the connectivity to the network through different physical ports, connector types and speed.

All the TOE models except OS9900 are stackable switches, where both the CMM software and NIM software execute on the same processor. The OS9900 switches are modular switches with multiple slots that house blade hardware components, where the CMM software and NIM software execute on different hardware blades. All the cryptographic functions used by the TOE in the evaluated configuration are implemented in the CMM software which runs on the processors listed in Table 1.

A switch in the OmniSwitch family can operate in two different modes: Standalone and Virtual Chassis (VC). A virtual chassis is a group of switches managed through a single management IP address that operates as a single switch and router. In the virtual chassis mode, two or more physical switches are connected through Virtual Fabric Links (VFL) and use a specific protocol to communicate each other.

The virtual chassis mode is not allowed in the evaluated configuration. The TOE is a physical and standalone network device in the evaluated configuration.

1.5.1 TOE Physical Boundary

The TOE physical boundary is the device chassis encompassing hardware and software. The TOE software is AOS software, which executes on the hardware platforms listed in the following table.

Table 1: TOE Hardware Platforms

Model	Switch Processor	Application Processor Series	Software Image	Network Interface
OmniSwitch 6360	Marvell 98DX236S	Arm ARMv7-A Cortex-A	Nosa.img	Marvell AlleyCat 3
	Marvell 98DX233S			
OmniSwitch 6465	Marvell 98DX3233	Arm ARMv7-A Cortex-A	Nos.img	Marvell AlleyCat 3
OmniSwitch 6465T	Marvell 98DX3233	Arm ARMv7-A Cortex-A	Nos.img	Marvell AlleyCat 3
OmniSwitch 6560	Marvell 88F6820	Arm ARMv7-A Cortex-A	Nos.img	Marvell AlleyCat 3
OmniSwitch 6570	Marvell 98DX3501	Arm ARMv8.2-A Cortex-A	Wos.img	Marvell AlleyCat 5
	Marvell 98DX3510			
OmniSwitch 6860N	Intel Atom C3338	Intel Goldmont Atom	Uosn.img	Broadcom Trident3
	Intel Atom C3558			
OmniSwitch 6865	Broadcom BCM56342	Arm ARMv7-A Cortex-A	Uos.img	Broadcom Helix4
OmniSwitch 6870	Intel Atom C3338	Intel Goldmont Atom	Kos.img	Marvell Prestera DX
	Intel Atom C3558			
OmniSwitch 6900	Intel Atom C3558	Intel Goldmont Atom	Yos.img	Broadcom Trident3
	Intel Xeon D1518	Intel Broadwell Xeon		
OmniSwitch 9900	Intel Atom C2518	Intel Silvermont Atom	Mos.img	Marvell Prestera DX
	Intel Atom C3558	Intel Goldmont Atom		

All the TOE models perform the same security functions with respect to this evaluation. The main distinctions are about physical characteristics, processor model, memory capacity, the type of network interfaces, the number of physical ports, and the supported network features (switching or routing).

The TOE also includes the following documentation providing information for installing, configuring, and maintaining the TOE in the evaluated configuration:

- Preparation and Operation of Common Criteria Evaluated OmniSwitch Products (NDcPP) – AOS Release 8.10.
  This document will hereafter be referred to as [CCGUIDE] throughout the ST.

1.5.2 TOE Logical Boundary

The TOE provides the following security functions in conformance to the protection profiles and packages listed in Section 2 “CC Conformance Claim”.

1.5.2.1 Security Audit

The TOE generates audit records for security-related events. The audit records contain time stamps and other information about the events, which are helpful to the administrator in investigating the issues related to the configuration and operation of the TOE.

The TOE writes audit records to a set of circular log files stored in the flash memory for permanent storage. These entries are tagged with the AOS application ID of the TOE subsystem that triggers the audit records to be generated. The TOE can be configured to transfer the audit data to an external syslog server in real-time using the TLS protocol.

The TOE allows administrators to configure the maximum size allowed for the log files. Once the files are full, the TOE overwrites the oldest records.

1.5.2.2 Cryptographic Support

The TOE implements the cryptography supporting the following functionalities:

- Establishment of secure network connections using the SSH and TLS protocols
- Validation of X.509 certificates
- Storage of user passwords
- Digital signature verification of TOE software updates

The TOE provides cryptographic support using the OpenSSL library version 3.0.13, which is bundled in the TOE. The TOE also includes a software-based, SP800-90B compliant Entropy Source seeding DRBGs with full entropy.

1.5.2.3 Identification and Authentication

Administrators can access the TOE through either the local console port or a remote SSH client. The TOE authenticates administrators before granting access to management functions. For local administrative sessions, authentication is performed using passwords. For remote sessions, administrators are authenticated via SSH passwords or SSH public keys. The TOE also supports validation of X.509 certificates for authenticating external IT entities.

The TOE provides administrator-configurable settings to enforce password complexity requirements. The TOE can be configured to disable an administrator account following a configurable number of failed remote login attempts using passwords.

1.5.2.4 Security Management

The TOE provides a Command-Line Interface (CLI) to the administrators for managing the configurable aspects of the TOE. The TOE provides a range of management functions, which are restricted to administrators with the appropriate privileges.

The TOE includes a built-in "admin" account with full privileges for all commands on the TOE. Additional configured administrator accounts can be created with varying levels of privileges. Both the built-in and configured administrator accounts are considered Security Administrator in this ST.

1.5.2.5 Protection of the TOE Security Functionality (TSF)

The TOE includes protective features to maintain the integrity and reliability of its security functions.

- The TOE uses the filesystem access control to protect sensitive data such as cryptographic keys and credentials.
- The TOE employs the digital signature to ensure trusted updates to the TOE software.
- The TOE performs self-tests to verify the correct operation of cryptographic services and the overall integrity of TOE software.

- The TOE provides a reliable system date and time, which is used for audit record time stamps, SSH rekeying, certificate validation, account lockout, and administrative session termination.

1.5.2.6 TOE Access

The TOE can be configured to display a login banner when an administrator establishes an interactive session (both local and remote). The TOE terminates interactive sessions after an administrator-defined period of inactivity. The TOE allows administrators to terminate their own administrative sessions.

1.5.2.7 Trusted Path/Channels

The TOE includes the OpenSSL library version 3.0.13 and OpenSSH package version 9.8. The TOE implements the TLS protocol version 1.2 (TLS v1.2) and the SSH protocol version 2 (SSHv2) using OpenSSL and OpenSSH, respectively.

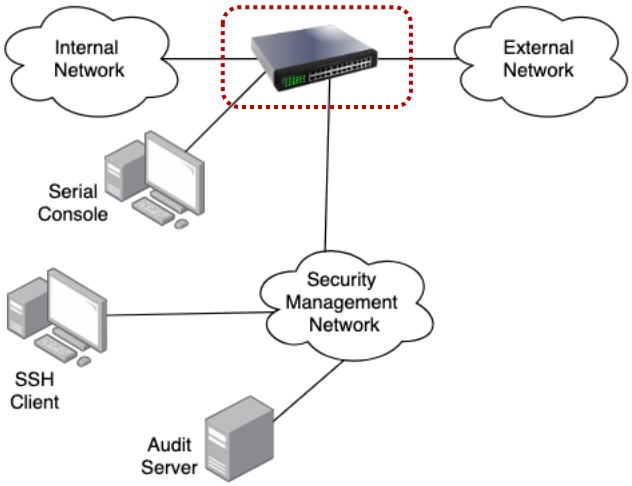
- The TOE uses the TLS protocol to secure the communication with the external audit server.
- The TOE uses the SSH protocol to protect the administrative sessions from the administrator's workstation to the TOE.

1.5.3 TOE Operational Environment

Figure 2 illustrates the TOE and its operational environment. The red dotted line encloses the physical boundary of the TOE.

- The TOE is located between the external and internal networks or within the internal networks of an organization. The TOE can be connected to external IT entities (e.g., syslog audit server).
- Administrators log into the TOE and perform management functions via a Command-Line Interface (CLI). These activities can be performed via a Serial Console connected to the TOE via a dedicated port, or using a Secure Shell (SSH) client from a computer over the network.

Figure 2: Operational Environment



The intended TOE environment is a secure data center where the TOE is protected from unauthorized physical access. Only security administrators have physical access to the hardware and are permitted to use the serial console. Appropriate administrator security policies and procedural guidance must be in place to govern the operational management of the TOE within its operational environment.

The TOE is not intended for use as a general-purpose system and executes only the services required to perform its intended functions.

1.5.4 Excluded TOE Features

The following features interfere with the TOE security functionality claims and must be either disabled or not configured for use in the evaluated configuration.

- File Transfer Protocol (FTP) access to the TOE
- Telnet access to the TOE
- Webview access to the TOE
- Hypertext Transfer Protocol (HTTP)
- Simple Network Management Protocol (SNMP)
- Lightweight Directory Access Protocol (LDAP) for external authentication
- Remote Authentication Dial-In User Service (RADIUS) for external authentication
- Network Time Protocol (NTP)
- Captive Portal
- Terminal Access Controller Access-Control System Plus (TACACS+)
- Port Mobility Rules

2 CC Conformance Claim

Common Criteria (CC) version 3.1 revision 5 is the basis for this conformance claim. This Security Target (ST) is CC Part 2 extended and CC Part 3 conformant.

This ST claims exact conformance to the following Protection Profiles (PPs) and Packages:

- collaborative Protection Profile for Network Devices, Version 3.0e, 2023-12-06.
  This document will hereafter be referred to as [CPP_ND_V3.0E] throughout the ST.
- Functional Package for Secure Shell (SSH), Version 1.0, 2021-05-13.
  This document will hereafter be referred to as [PKG_SSH_V1.0] throughout the ST.

Table 2 lists the NIAP Technical Decisions (TDs) for [CPP_ND_V3.0E] at the time of the evaluation and a statement of applicability to the evaluation.

Table 2: NIAP Technical Decisions for [CPP_ND_V3.0E]

TD #	Description	Applicable?	Non-applicability Rationale
TD0923	NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Yes	
TD0921	NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Yes	
TD0900	NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Yes	
TD0899	NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	Yes	
TD0886	Clarification to FAU_STG_EXT.1 Test 6	Yes	
TD0880	NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Yes	
TD0879	NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Yes	
TD0868	NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	No	The ST does not claim FCS_IPSEC_EXT.1.
TD0836	NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Yes	

Table 3 contains the NIAP Technical Decisions (TDs) for [PKG_SSH_V1.0] at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP Technical Decisions for [PKG_SSH_V1.0]

TD #	Description	Applicable?	Non-applicability Rationale
TD0967	Allowance of Kex-strict in PKG_SSH_V1.0	Yes	
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Yes	

TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update	Yes	
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package	Yes	
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	

3 Security Problem Definition

The security problem definition is taken directly from [CPP_ND_V3.0E].

3.1 Threats

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

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

T.WEAK_CRYPTOGRAPHY

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

T.UNTRUSTED_COMMUNICATION_CHANNELS

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

T.WEAK_AUTHENTICATION_ENDPOINTS

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

T.UPDATE_COMPROMISE

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

T.UNDETECTED_ACTIVITY

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

T.SECURITY_FUNCTIONALITY_COMPROMISE

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

Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.

T.SECURITY_FUNCTIONALITY_FAILURE

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

3.2 Assumptions

A.PHYSICAL_PROTECTION

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

A.LIMITED_FUNCTIONALITY

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

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

A.NO_THRU_TRAFFIC_PROTECTION

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

A.TRUSTED_ADMINISTRATOR

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

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

A.REGULAR_UPDATES

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

A.ADMIN_CREDENTIALS_SECURE

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

A.RESIDUAL_INFORMATION

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

3.3 Organizational Security Policies

P.ACCESS_BANNER

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

4 Security Objectives

The security objectives are taken directly from [CPP_ND_V3.0E].

4.1 Objectives for the TOE

[CPP_ND_V3.0E] does not define security objectives for the TOE.

4.2 Objectives for the Operational Environment

OE.PHYSICAL

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

OE.NO_GENERAL_PURPOSE

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

OE.NO_THRU_TRAFFIC_PROTECTION

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

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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

OE.UPDATES

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

OE.ADMIN_CREDENTIALS_SECURE

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

OE.RESIDUAL_INFORMATION

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

4.3 Security Objectives Rationale

The security objectives rationale is defined in [CPP_ND_V3.0E].

5 Extended Components Definition

This Security Target claims exact conformance to [CPP_ND_V3.0E] and [PKG_SSH_V1.0], therefore all extended requirements in the Security Target are drawn from those documents.

The extended requirement components are defined at the following locations:

- [CPP_ND_V3.0E]: Appendix C.
- [PKG_SSH_V1.0]: Section 3 and Appendix B.

6 Security Requirements

The following conventions are used to indicate the operations performed within the ST on security requirement components:

- Selections are shown in **bold text** and are surrounded by square brackets.
- Assignments are shown in *italic text* and are surrounded by square brackets. The assignments within a selection are shown in ***bold italics***.
- Iterations are identified by appending a suffix to the original SFR.
- Refinements added to the text are shown in underlined text, deletions are shown as ~~strikethrough text~~.

6.1 TOE Security Functional Requirements

The Security Functional Requirements (SFRs) are drawn from [CPP_ND_V3.0E] and [PKG_SSH_V1.0]. The table below summarizes the SFRs satisfied by the TOE. The table includes four columns:

- Column 1 identifies the SFR class.
- Column 2 lists the SFR component.
- Column 3 indicates the PP or Package which the SFR is drawn from.
- Column 4 specifies the SFR type: Mandatory (M), Optional (O), or Selection-based (S).

Table 4: Security Functional Requirements

Requirement Class	Requirement Components	Source	Type
Security Audit (FAU)	FAU_GEN.1	[CPP_ND_V3.0E]	M
	FAU_GEN.2	[CPP_ND_V3.0E]	M
	FAU_STG.1	[CPP_ND_V3.0E]	O
	FAU_STG_EXT.1	[CPP_ND_V3.0E]	M
	FAU_STG_EXT.3	[CPP_ND_V3.0E]	O
Cryptographic Support (FCS)	FCS_CKM.1	[CPP_ND_V3.0E]	M
	FCS_CKM.2	[CPP_ND_V3.0E]	M
	FCS_CKM.4	[CPP_ND_V3.0E]	M
	FCS_COP.1/DataEncryption	[CPP_ND_V3.0E]	M
	FCS_COP.1/SigGen	[CPP_ND_V3.0E]	M
	FCS_COP.1/Hash	[CPP_ND_V3.0E]	M
	FCS_COP.1/KeyedHash	[CPP_ND_V3.0E]	M
	FCS_RBG_EXT.1	[CPP_ND_V3.0E]	M
	FCS_SSH_EXT.1	[PKG_SSH_V1.0]	M
	FCS_SSHS_EXT.1	[PKG_SSH_V1.0]	S

Requirement Class	Requirement Components	Source	Type
	FCS_TLSC_EXT.1	[CPP_ND_V3.0E]	S
	FCS_TLSC_EXT.2	[CPP_ND_V3.0E]	O
Identification and Authentication (FIA)	FIA_AFL.1	[CPP_ND_V3.0E]	S
	FIA_PMG_EXT.1	[CPP_ND_V3.0E]	S
	FIA_UAU.7	[CPP_ND_V3.0E]	S
	FIA_UIA_EXT.1	[CPP_ND_V3.0E]	M
	FIA_X509_EXT.1/Rev	[CPP_ND_V3.0E]	S
	FIA_X509_EXT.2	[CPP_ND_V3.0E]	S
	FIA_X509_EXT.3	[CPP_ND_V3.0E]	S
Security Management (FMT)	FMT_MOF.1/Functions	[CPP_ND_V3.0E]	S
	FMT_MOF.1/ManualUpdate	[CPP_ND_V3.0E]	M
	FMT_MTD.1/CoreData	[CPP_ND_V3.0E]	M
	FMT_MTD.1/CryptoKeys	[CPP_ND_V3.0E]	S
	FMT_SMF.1	[CPP_ND_V3.0E]	M
	FMT_SMR.2	[CPP_ND_V3.0E]	M
Protection of the TSF (FPT)	FPT_APW_EXT.1	[CPP_ND_V3.0E]	S
	FPT_SKP_EXT.1	[CPP_ND_V3.0E]	M
	FPT_STM_EXT.1	[CPP_ND_V3.0E]	M
	FPT_TST_EXT.1	[CPP_ND_V3.0E]	M
	FPT_TUD_EXT.1	[CPP_ND_V3.0E]	M
TOE Access (FTA)	FTA_SSL.3	[CPP_ND_V3.0E]	M
	FTA_SSL.4	[CPP_ND_V3.0E]	M
	FTA_SSL_EXT.1	[CPP_ND_V3.0E]	S
	FTA_TAB.1	[CPP_ND_V3.0E]	M
Trusted Path/Channels (FTP)	FTP_ITC.1	[CPP_ND_V3.0E]	M
	FTP_TRP.1/Admin	[CPP_ND_V3.0E]	M

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

Origin: [CPP_ND_V3.0E]

Applied TDs: TD0777

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 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 ~~Table 2~~.

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 ~~Table 2~~.

Table 5: Auditable Events

Requirement	Auditable Event	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FAU_STG_EXT.3	Low storage space for audit events.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.

Requirement	Auditable Event	Additional Audit Record Contents
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	• [**Failure to establish SSH Connection**] • [**Establishment of SSH connection**] • [**Termination of SSH connection session**] • [**Dropping of packet(s) outside defined size limits**]	• [**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)**] • [**Packet size**]
FCS_SSHS_EXT.1	No event specified	None
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	None	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanisms	Origin of the attempt (e.g., IP address)
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE’s trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE’s trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.

Requirement	Auditable Event	Additional Audit Record Contents
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session lock	None.
FTA_TAB.1	• None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	• None • None • Reason for failure
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

6.1.1.2 FAU_GEN.2 User Identity Association

Origin: [CPP_ND_V3.0E]

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

6.1.1.3 FAU_STG.1 Protected Audit Trail Storage

Origin: [CPP_ND_V3.0E]

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

6.1.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

Origin: [CPP_ND_V3.0E]

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself. In addition [

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

FAU_STG_EXT.1.3 The TSF shall maintain a [**log file, [*a set of circular log files*]**] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4 The TSF shall be able to store [**persistent**] audit records locally within a minimum storage size of [*125 KB*] .

FAU_STG_EXT.1.5 The TSF shall [**overwrite previous audit records according to the following rule: [*overwrite the data present in the oldest log file*]**] when the local storage space for audit data is full.

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records [**manual export, ability to view locally**].

6.1.1.5 FAU_STG_EXT.3 Action in Case of Possible Audit Data Loss

Origin: [CPP_ND_V3.0E]

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

Origin: [CPP_ND_V3.0E]

Applied TDs: TD0921

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, 3072 bits*] 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, P-521] 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].**

].

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment

Origin: [CPP_ND_V3.0E]

FCS_CKM.2.1 The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- **RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2”;**
- **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].**

].

6.1.2.3 FCS_CKM.4 Cryptographic Key Destruction

Origin: [CPP_ND_V3.0E]

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 [zeros]**];
- 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 [
  - **instructs a part of the TSF to destroy the abstraction that represents the key**]

that meets the following: No Standard.

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

Origin: [CPP_ND_V3.0E]

FCS_COP.1.1/ DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [**CBC, CTR, GCM**] modes and cryptographic key sizes [**128 bits, 256 bits**] that meet the following: AES as specified in ISO 18033-3, [**CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772**].

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

Origin: [CPP_ND_V3.0E]

Applied TDs: TD0921

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: [*2048, 3072 bits*],**
- **For ECDSA: [*P-256, P-384, P-521*]**

]

that meets 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, P-521] 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.**

].

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

Origin: [CPP_ND_V3.0E]

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.

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

Origin: [CPP_ND_V3.0E]

FCS_COP.1.1/ KeyedHash The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [**HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key sizes [*160, 256, 384, 512 bits*] and message digest sizes [**160, 256, 384, 512**] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

6.1.2.8 FCS_RBG_EXT.1 Random Bit Generation

Origin: [CPP_ND_V3.0E]

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

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

6.1.2.9 FCS_SSH_EXT.1 SSH Protocol

Origin: [PKG_SSH_V1.0]

Applied TDs: TD0909

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, [**4256, 4344, 5647, 5656, 6668, 8268, 8308, 8332**] and no other standard.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- **“password” (RFC 4252),**
- **“keyboard-interactive” (RFC4256),**
- **“publickey” (RFC 4252): [**
  - **rsa-sha2-256 (RFC 8332),**
  - **rsa-sha2-512 (RFC 8332),**
  - **ecdsa-sha2-nistp256 (RFC 5656),**
  - **ecdsa-sha2-nistp384 (RFC 5656),**
  - **ecdsa-sha2-nistp521 (RFC 5656),**

  ]

] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256 KB (262,144 bytes)*] in an SSH transport connection are dropped.

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),**
- **aes128-gcm@openssh.com (RFC 5647),**
- **aes256-gcm@openssh.com (RFC 5647)**

] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [

- **hmac-sha2-256 (RFC 6668),**
- **hmac-sha2-512 (RFC 6668),**
- **implicit**

] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [

- **diffie-hellman-group14-sha256 (RFC 8268),**
- **diffie-hellman-group16-sha512 (RFC 8268),**
- **ecdh-sha2-nistp256 (RFC 5656),**
- **ecdh-sha2-nistp384 (RFC 5656),**
- **ecdh-sha2-nistp521 (RFC 5656),**

] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in [

- **RFC 4253 (Section 7.2),**
- **RFC 5656 (Section 4)**

] to derive the following cryptographic keys from a shared secret: session keys.

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.

6.1.2.10 FCS_SSHS_EXT.1 SSH Protocol - Server

Origin: [PKG_SSH_V1.0]

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),**

- **ecdsa-sha2-nistp384 (RFC 5656),**
- **ecdsa-sha2-nistp521 (RFC 5656)**

].

6.1.2.11 FCS_TLSC_EXT.1 TLS Client Protocol

Origin: [CPP_ND_V3.0E]

FCS_TLSC_EXT.1.1 The TSF shall implement [**TLS 1.2 (RFC 5246)**] supporting the following ciphersuites: [

- **TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268**
- **TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268**
- **TLS_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246**
- **TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246**
- **TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288**
- **TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288**
- **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289**
- **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289**
- **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289**
- **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**
- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289**
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**
- **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289**
- **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289**

] and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [**the reference identifier per RFC 6125 Section 6**].

FCS_TLSC_EXT.1.3 The TSF shall not establish a trusted channel if the server certificate is invalid [

- **without any administrator overwrite mechanism.**

].

FCS_TLSC_EXT.1.4 The TSF shall [**present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1] and no other curves/groups**] in the Client Hello.

FCS_TLSC_EXT.1.5 The TSF shall [

- **present the signature_algorithms extension with support for the following algorithms: [**
  - **rsa_pkcs1 with sha256(0x0401),**
  - **rsa_pkcs1 with sha384(0x0501),**
  - **rsa_pkcs1 with sha512(0x0601),**
  - **ecdsa_secp256r1 with sha256(0x0403),**
  - **ecdsa_secp384r1 with sha384(0x0503),**
  - **ecdsa_secp521r1 with sha512(0x0603),**
  - **rsa_pss_rsae with sha256(0x0804),**
  - **rsa_pss_rsae with sha384(0x0805),**
  - **rsa_pss_rsae with sha512(0x0806),**
  - **rsa_pss_pss with sha256(0x0809),**
  - **rsa_pss_pss with sha384(0x080a),**
  - **rsa_pss_pss with sha512(0x080b)**

  ] and no other algorithms;

].

FCS_TLSC_EXT.1.6 The TSF [**provides**] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7 The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8 The TSF shall [**not use PSKs**].

FCS_TLSC_EXT.1.9 The TSF shall [**reject [TLS 1.2] renegotiation attempts**].

6.1.2.12 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

Origin: [CPP_ND_V3.0E]

FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management

Origin: [CPP_ND_V3.0E]

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 to 999*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [**prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [*unlocking account action*] is taken by an Administrator;**

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].

6.1.3.2 FIA_PMG_EXT.1 Password Management

Origin: [CPP_ND_V3.0E]

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 [*1*] and [*30*] characters.

6.1.3.3 FIA_UAU.7 Protected Authentication Feedback

Origin: [CPP_ND_V3.0E]

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

6.1.3.4 FIA_UIA_EXT.1 User Identification and Authentication

Origin: [CPP_ND_V3.0E]

Applied TDs: [TD0900]

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**].

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.

FIA_UIA_EXT.1.3 The TSF shall provide the following remote authentication mechanisms [**SSH password, SSH public key**] and [**no other mechanism**]. The TSF shall provide the following local authentication mechanisms [**password-based**].

FIA_UIA_EXT.1.4 The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

6.1.3.5 FIA_X509_EXT.1/Rev X.509 Certificate Validation

Origin: [CPP_ND_V3.0E]

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [**the Online Certificate Status Protocol (OCSP) as specified in RFC 6960**].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

6.1.3.6 FIA_X509_EXT.2 X.509 Certificate Authentication

Origin: [CPP_ND_V3.0E]

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

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [**not accept the certificate**].

6.1.3.7 FIA_X509_EXT.3 X.509 Certificate Requests

Origin: [CPP_ND_V3.0E]

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

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

Origin: [CPP_ND_V3.0E]

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, handling of audit data**] to Security Administrators.

6.1.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

Origin: [CPP_ND_V3.0E]

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

6.1.4.3 FMT_MTD.1/CoreData Management of TSF Data

Origin: [CPP_ND_V3.0E]

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

6.1.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data

Origin: [CPP_ND_V3.0E]

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

6.1.4.5 FMT_SMF.1 Specification of Management Functions

Origin: [CPP_ND_V3.0E]

Applied TDs: TD0880

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 configure local audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size);**
  - **Ability to manage the cryptographic keys;**
  - **Ability to configure the list of supported (D)TLS ciphers;**
  - **Ability to re-enable an Administrator account;**

- **Ability to set the time which is used for time-stamps;**
- **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;**

].

6.1.4.6 FMT_SMR.2 Restrictions on Security Roles

Origin: [CPP_ND_V3.0E]

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

6.1.5 Protection of TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

Origin: [CPP_ND_V3.0E]

FPT_APW_EXT.1.1 The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext administrative passwords.

6.1.5.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

Origin: [CPP_ND_V3.0E]

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

6.1.5.3 FPT_STM_EXT.1 Reliable Time Stamps

Origin: [CPP_ND_V3.0E]

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

FPT_STM_EXT.1.2 The TSF shall [**allow the Security Administrator to set the time**].

6.1.5.4 FPT_TST_EXT.1 TSF Testing

Origin: [CPP_ND_V3.0E]

Applied TDs: TD0836

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;
- [**no other**] self-tests [**none**].

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2 The TSF shall respond to [**all failures**] by [**rebooting**].

6.1.5.5 FPT_TUD_EXT.1 Trusted Update

Origin: [CPP_ND_V3.0E]

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**].

FPT_TUD_EXT.1.2 The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [**no other update mechanism**].

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

6.1.6 TOE Access (FTA)

6.1.6.1 FTA_SSL.3 TSF-initiated Termination

Origin: [CPP_ND_V3.0E]

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

6.1.6.2 FTA_SSL.4 User-initiated Termination

Origin: [CPP_ND_V3.0E]

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

6.1.6.3 FTA_SSL_EXT.1 TSF-initiated Session Locking

Origin: [CPP_ND_V3.0E]

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [

- **terminate the session**]

after a Security Administrator-specified time period of inactivity.

6.1.6.4 FTA_TAB.1 Default TOE Access Banners

Origin: [CPP_ND_V3.0E]

FTA_TAB.1.1 Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

6.1.7 Trusted Path/Channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

Origin: [CPP_ND_V3.0E]

FTP_ITC.1.1 The TSF shall be capable of using [**TLS**] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [**no other capabilities**] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2 The TSF shall permit [**the TSF**] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*sending audit records to the audit server*].

6.1.7.2 FTP_TRP.1/Admin Trusted Path

Origin: [CPP_ND_V3.0E]

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

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

FTP_TRP.1.3/Admin The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

6.2 Security Functional Requirements Rationale

The SFR rationale is defined in [CPP_ND_V3.0E].

6.3 Security Assurance Requirements

The Security Assurance Requirements (SARs) are included by reference from [CPP_ND_V3.0E]. The table below summarizes the SARs for TOE evaluation. The ST includes all the mandatory SARs and an optional SAR (i.e., ALC_FLR.2 Flaw reporting procedures).

Table 6: Security Assurance Requirements

Requirement Class	Requirement Components

Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
	Flaw reporting procedures (ALC_FLR.2)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

6.4 Security Assurance Requirements Rationale

The SAR rationale is defined in [CPP_ND_V3.0E].

7 TOE Summary Specification

As per [CPP_ND_V3.0E] and [PKG_SSH_V1.0], the TOE supports the following major security features:

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

7.1 Security Audit

The TOE is a standalone device that can store audit data both locally in non-volatile memory and remotely on a syslog server. Audit functionality in the TOE is provided via the switch logging feature, which records audit events for all administrative operations performed.

The TOE supports the severity levels detailed in the following table. Level 6 (Info) is enabled for all events by default and is the minimum severity level required in the evaluated configuration.

Table 7: TOE Audit Record Levels

Severity Level	Type	Description
1 (highest severity)	Alarm	A serious error has occurred. The system is about to crash and reboot.
2	Error	System functionality is reduced.
3	Alert	A violation has occurred.
4	Warning	An unexpected, non-critical event has occurred
5	Event	A clear readable customer event.
6 (default)	Info	Any other non-debug message.
7	Debug1	A normal event debug message.
8	Debug2	A debug-specific message.
9 (lowest severity)	Debug3	A maximum verbosity debug message.

When an audit event request is made, the severity level on the request is compared to the severity level assigned to the application ID for which the event occurs. If the severity level number of the log request is less than or equal to that of the application ID, the log message is generated and placed in the log file.

Specific security and administrative events that are required to be audited by this ST are generated with Level 6 (Info) and their description prefixed with the "EVENT-AUDIT". The administrator may configure the severity level either globally for all applications or on a per application basis using the “swlog appid” command.

7.1.1 FAU_GEN.1

Each audit record contains the date and time of the event, type of event, subject identity (whenever possible) and outcome (success or failure). The type of event and outcome are included in the Log Message field which specifies the condition recorded.

For the events of managing cryptographic keys (e.g., generation, import, update, deletion, view and verification of certificates/keys), the audit record includes the full "aaa certificate" command line entered by the administrator, which provides the certificates and/or keys involved in the operation, and whether the operation succeeded or failed. The audit record also includes the description of the error in case of a failure.

7.1.2 FAU_GEN.2

The TSS requirement for FAU_GEN.2 is covered by the TSS requirement for FAU_GEN.1.

7.1.3 FAU_STG.1

Please refer to TSS section for FAU_STG_EXT.1.

7.1.4 FAU_STG_EXT.1

The TOE is a standalone device that can be configured to store audit records in the local filesystem and to send them to a remote syslog server. A secure communication channel is established between the TOE and the external audit server using the TLS protocol version 1.2 to protect audit data during transmission.

The audit record is sent to the syslog server in real-time, immediately after the event occurs. If communication fails, the audit record is kept locally and not resent to the syslog server. The TOE will attempt to reconnect to the audit server at fixed intervals (every 60 seconds) or whenever a new audit generation request is received.

For local logging, the TOE stores audit data in a set of log files which are persistently located in the /flash directory and prefixed as "swlog_". When the active log file reaches its size limit, the TOE overwrites the oldest file in the log set using circular logging. The maximum size of all log files is the same and can be configured using the “swlog output flash-file-size <kilobytes>” command. The allowed values for the maximum log file size range from 125 to 12500 kilobytes. By default, the log file size is set to 1250 kilobytes.

The following table summarizes the information about the log file set. For example, when the active log file (i.e., swlog) reaches its size limit, the TOE will first rename each existing archived log file in the circular set from swlog.(i) to swlog.(i+1). The oldest archived log file (i.e., swlog.6) is removed in case the set is full. Next, the TOE will close and rename the swlog file as the latest archived file (i.e., swlog.0). Lastly, the TOE will create a new and empty active log file.

Table 8: Audit Local Storage

Log files	File names	Parameter	Default size	Allowed size
Eight files	swlog_<suffix>, swlog_<suffix>.0 through swlog_<suffix>.6	Maximum size for each file (in KB)	1250 KB	125 KB to 12500 KB

The TOE protects log files from unauthorized modification or deletion by enforcing access control to the commands used to manipulate log files. Only administrators have the access to the commands for changing or clearing the contents of log files.

7.1.5 FAU_STG_EXT.3

The TOE notifies the administrator when the utilized space of a log file reaches a configured threshold percentage of the maximum file size. The warning message is appended to the log file.

7.2 Cryptographic Support

The TOE implements cryptographic protocols and algorithms using the following cryptographic libraries:

- OpenSSL version 3.0.13 for the TLS v1.2 protocol and cryptographic algorithm support.
- OpenSSH version 9.8 for the SSHv2 protocol. OpenSSH uses OpenSSL for the underlying cryptographic algorithms.

The following table summarizes the cryptographic services used by the TOE, describing the cryptographic operations, algorithms, and applicable standards. The table also includes the Cryptographic Algorithm Validation Program (CAVP) certificate number for each of the cryptographic algorithms.

Table 9: Mapping of SFRs to CAVP certificates (OpenSSL cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_CKM.1	RSA	Modulus: 2048, 3072 bits	[FIPS186-5]	A6866, A6871
	ECC	Curves: P-256, P-384, P-521	[FIPS186-5]	A6866, A6871
	FFC	Groups: MODP-2048, MODP-4096	[SP800-56Ar3]	A6857
FCS_CKM.2	RSA Key Establishment	Modulus: 2048, 3072 bits	[RFC8017]	CCTL tested
	KAS-ECC-SSC	Curves: P-256, P-384, P-521	[SP800-56Ar3]	A6866, A6871
	KAS-FFC-SSC	Groups: MODP-2048, MODP-4096	[SP800-56Ar3]	A6857
FCS_COP.1/Data Encryption	AES-CBC	128 bits, 256 bits encrypt, decrypt	[SP800-38A]	A6847, A6869
	AES-CTR	128 bits, 256 bits encrypt, decrypt	[SP800-38A]	A6847, A6869
	AES-GCM	128 bits, 256 bits encrypt, decrypt	[SP800-38D]	A6851, A6858
FCS_COP.1/SigGen	RSA SigGen	Modulus: 2048, 3072 bits Hash: SHA2-256, SHA2-384, SHA2-512 Padding: PKCS#1 v1.5 and PSS	[FIPS186-5]	A6866, A6871
	RSA SigVer	Modulus: 2048, 3072 bits Hash: SHA2-256, SHA2-384, SHA2-512 Padding: PKCS#1 v1.5 and PSS	[FIPS186-5]	A6866, A6871

SFR	Algorithm	Capabilities	Standard	CAVP cert.
	ECDSA SigGen	Curves: P-256, P-384, P-521 Hash: SHA2-256, SHA2-384, SHA2-512	[FIPS186-5]	A6866, A6871
	ECDSA SigVer	Curves: P-256, P-384, P-521 Hash: SHA2-256, SHA2-384, SHA2-512	[FIPS186-5]	A6866, A6871
FCS_COP.1/Hash	SHA-1, SHA2-256, SHA2-384, SHA2-512	Byte-oriented mode	[FIPS180-4]	A6866, A6871
FCS_COP.1/KeyedHash	HMAC-SHA-1	Key size: 160 bits Block size: 512 bits MAC size: 160 bits	[FIPS198-1]	A6866, A6871
	HMAC-SHA-256	Key size: 256 bits Block size: 512 bits MAC size: 256 bits	[FIPS198-1]	A6866, A6871
	HMAC-SHA-384	Key size: 384 bits Block size: 1024 bits MAC size: 384 bits	[FIPS198-1]	A6866, A6871
	HMAC-SHA-512	Key size: 512 bits Block size: 1024 bits MAC size: 512 bits	[FIPS198-1]	A6866, A6871
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A6844

7.2.1 FCS_CKM.1

Please refer to the CAVP table above for the asymmetric key generation schemes and key sizes supported by the TOE.

As the TLS client, the TOE uses RSA and ECC key generation schemes for TLS client certificate provisioning. The TOE generates ephemeral ECC key pairs during the key exchange phase of a TLS session.

As the SSH server, the TOE supports generating SSH host key pairs using RSA and ECC schemes. The TOE generates ephemeral ECC and FFC keys for key exchange during SSH sessions.

7.2.2 FCS_CKM.2

Please refer to the CAVP table above for the key establishment schemes and key sizes supported by the TOE.

The TOE uses RSA and ECC-based key exchange mechanisms to establish a shared secret in a TLS session. The TOE uses the ECC-based and FFC-based key exchange mechanisms for SSH sessions.

7.2.3 FCS_CKM.4

All cryptographic keys residing in volatile storage are maintained in plaintext format. To delete a key in volatile storage, the TOE overwrites it with zeroes and releases the allocated memory when the key is no longer needed (e.g., the cryptographic handle is freed or a TLS session is finished).

For destruction of a key in non-volatile storage, the TOE deletes the file containing that key from the filesystem, either automatically through filesystem APIs when the key is no longer needed or upon the requests from administrators issuing the relevant CLI commands. The filesystem API function remove() is used to delete the code signing certificate file. The CLI command "AAA certificate delete" is used to delete the TOE TLS client certificate files and the "rm" command is for deleting other files.

The following table shows the cryptographic keys used by the TOE, storage locations (including the filenames in case of non-volatile storage), storage formats, and destruction methods.

Table 10: Storage and Destruction of Cryptographic Keys

Certificate/Key	Storage Location	Storage Format	Destruction Method
SSH host key (public and private key pair)	/flash/system/ssh_host_rsa_key /flash/system/ssh_host_rsa_key.pub /flash/system/ssh_host_ecdsa_key /flash/system/ssh_host_ecdsa_key.pub	Plaintext	Removal through CLI commands
SSH user public key	/flash/system/<username>_rsa.pub /flash/system/<username>_ecdsa.pub	Plaintext	Removal through CLI commands
SSH session key (shared secrets, encryption keys, MAC keys)	Volatile memory	Plaintext	Zeroization and deallocation when the session terminates
TOE TLS client certificate (public and private key pair)	flash/switch/cert.d/myCliCert.pem (or .crt) /flash/switch/cert.d/myCliPrivate.key	Plaintext	Removal through CLI commands
TLS server certificate (public key)	Volatile memory (received from the external audit server)	Plaintext	Zeroization and deallocation when the session terminates
TLS session key (shared secrets, encryption keys, MAC keys)	Volatile memory	Plaintext	Zeroization and deallocation when the session terminates
Code signing certificate (public key)	/tmp/*.pem (extracted from TOE software image during updating)	Plaintext	Removal through filesystem APIs after verifying the digital signature
CA root certificate for validating the code signing certificate (public key)	/etc/code_sign/ca.pem	Plaintext	Removal through CLI commands
CA root certificate for validating other certificates (public key)	/flash/switch/ca.d/certs.pem (CA bundle file)	Plaintext	Removal through CLI commands

7.2.4 FCS_COP.1/DataEncryption

Please refer to the CAVP table above for the encryption algorithms and key sizes supported by the TOE.

7.2.5 FCS_COP.1/SigGen

Please refer to the CAVP table above for the digital signature algorithms and key sizes supported by the TOE.

7.2.6 FCS_COP.1/Hash

Please refer to the CAVP table above for the hash algorithms supported by the TOE.

The TOE uses those hash algorithms for the following purposes: TOE software integrity verification, HMAC computation, digital signature operations, pseudorandom function (PRF) for key derivation in SSH, and user password protection.

7.2.7 FCS_COP.1/KeyedHash

The TOE implements HMAC keyed hash algorithms with SHA-1, SHA2-256, SHA2-384, and SHA2-512. Please refer to the CAVP table above for the parameters of the keyed hash algorithms.

7.2.8 FCS_RBG_EXT.1

The TOE uses the Deterministic Random Bit Generator (DRBG) implemented in OpenSSL. The DRBG is the CTR_DRBG based on the AES-256 algorithm.

The TOE seeds the DRBG using the OmniSwitch AOS Entropy Source, which has an entropy rate of one bit of entropy per bit (full entropy). The DRBG is seeded with 384 bits of entropy at initialization, and subsequently reseeded with 256 bits of entropy.

7.2.9 FCS_SSH_EXT.1

As the SSH server, the TOE supports the following user authentication methods:

- password: The username and password.
- keyboard-interactive: Password authentication is used over keyboard-interactive. The password is the only credential verified by the TOE.
- publickey: The SSH user key pair (RSA or ECDSA) is generated at the administrator workstation. The SSH user public key is registered on the TOE.

The TOE limits the size of SSH packets to 256K (256 × 1024) bytes and any packet greater than this size in an SSH transport connection will be dropped.

The following table outlines the algorithms used for the different usages of the SSH protocol implemented by the TOE.

Table 11: SSH Algorithms Supported by the TOE

Usage	Cryptographic Algorithm
Peer authentication (authenticating SSH server)	Public-key based using RSA and ECDSA (see public key algorithms)
User authentication (authenticating SSH client user)	Public-key based using RSA and ECDSA (see public key algorithms)
	Password-based using SHA-256
Key establishment	Elliptic Curve Diffie-Hellman with SHA-256, SHA-384 and SHA-512, and NIST curves P-256, P-384 and P-521

	Diffie-Hellman group 14 with SHA-256 and group 16 with SHA-512
Key derivation	SHA-256, SHA-384, and SHA-512, according to RFC 4253 (Section 7.2) and RFC 5656 (Section 4)
Confidentiality	AES (CBC and CTR modes) with 128-bit and 256-bit keys
Data integrity	HMAC-SHA2-256, HMAC-SHA2-512
Confidentiality and Data integrity	AES (GCM mode) with 128-bit and 256-bit keys (aes128-gcm@openssh.com and aes256-gcm@openssh.com)
Public key algorithms	RSAPKCS#1v1.5 with SHA2-256 and SHA2-512 (rsa-sha2-256 or rsa-sha2-512), using 2048-bit and 3072-bit keys
	ECDSA with SHA2-256, SHA2-384 and SHA2-512, and NIST curves P-256, P-384 and P-521

The TOE performs rekey of session keys when the data transmission threshold, which is 1GB, is surpassed, or if the session time limit, which is one hour, has passed.

7.2.10 FCS_SSHS_EXT.1

There is no TSS requirement for this SFR.

7.2.11 FCS_TLSC_EXT.1

The TOE implements the TLS protocol v1.2 using OpenSSL. Acting as the TLS client, the TOE enables the following TLS v1.2 ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The ciphersuites are selected in the order shown in the above list. In addition, the administrator can configure the ciphersuites that the TOE supports via management functions (e.g., ssl cipher CLI command).

The TOE performs certificate path validation of the server certificate during the TLS handshake. If the certificate cannot be successfully validated, e.g., it has expired or has been revoked, the TLS session is not established.

The configured server hostname for the external audit (TLS) server is used as the reference identifier for validating certificates. The TOE verifies that the TLS server's presented identifier matches the reference identifier using the following approach:

- If the server certificate includes a Domain Name System (DNS) domain name in the Subject Alternative Name (SAN) field, the TOE verifies that the DNS name matches the server hostname, according to [RFC6125].
- If the server certificate does not include a DNS identifier in the SAN field, the TOE verifies that the Common Name (CN) in the Subject field is a DNS name and matches the server hostname, according to [RFC6125].

If either of the verification methods above succeeds, then the certificate is trusted. Otherwise, the TLS session is not established.

The TOE does not support the use of wildcards or IP addresses in certificates.

The TOE implements the Supported Elliptic Curves Extension according to [RFC4492] with NIST curves secp256r1, secp384r1, and secp521r1. This behavior is performed by default and there is no security management function to configure it.

Key agreement parameters used for the server key exchange by ECDHE ciphersuites are determined based on the chosen elliptic curves negotiated between the TOE and TLS server from the list of NIST curves (secp256r1, secp384r1, and secp521r1).

The TOE implements the signature_algorithms extension with support for the following algorithms. This behavior is performed by default and there is no security management function to configure it.

- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1with sha384(0x0501),
- rsa_pkcs1 with sha512(0x0601),
- ecdsa_secp256r1 with sha256(0x0403),
- ecdsa_secp384r1 with sha384(0x0503),
- ecdsa_secp521r1 with sha512(0x0603),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),
- rsa_pss_pss with sha256(0x0809),
- rsa_pss_pss with sha384(0x080a),
- rsa_pss_pss with sha512(0x080b).

7.2.12 FCS_TLSC_EXT.2

The TOE supports mutual authentication when acting as the TLS client using X.509 certificates. The TOE is capable of providing a certificate in the TLS negotiation when the TLS server requests a certificate.

For mutual authentication, the TOE sends its client certificate upon the server's request. The TOE looks for the certificate and its private key in the certificate directory (/flash/switch/cert.d). Table 10 lists the naming conventions for certificates used for communication with external entities.

7.3 Identification and Authentication

The TOE provides local and remote access to administrators. Local access is provided through a serial console which is connected to the console port of the TOE. Remote access is provided through an SSH client program running on a remote workstation. Administrators must be identified and authenticated to the TOE via the local console or SSH client prior to allowing access to management functions.

7.3.1 FIA_AFL.1

The TOE manages authentication failure according to the following settings:

- Lockout window: the length of time a failed login attempt is aged before it is no longer counted as a failed attempt.
- Lockout threshold: the number of failed login attempts allowed within a given lockout window period of time.
- Lockout duration: the length of time a user account remains locked until it is automatically unlocked.

The lockout window is basically a moving observation window of time in which failed login attempts are counted. If a failed login attempt ages beyond the observation window of time, that attempt is no longer counted towards the threshold number. If the lockout window is set to 0, there is no observation window and failed login attempts are never aged out.

For password-based authentication, the TOE hashes the password entered by the user and compares the resulting hash value with the corresponding stored reference value. If the values do not match, the authentication failure count is incremented. In addition, the authentication failure count is decremented when a failed login attempt ages out. When the authentication failure count reaches the lockout threshold, the user account is locked out of the TOE for the lockout duration.

The account is unlocked when either of the following conditions is met. In either case, the authentication failure count is reset.

- The lockout duration expires.
- An administrator unlocks the user account via the CLI command.

The TOE includes a built-in "admin" account that can log in either at the console or via SSH. The “admin” account is protected from lockout; therefore, it is always available.

7.3.2 FIA_PMG_EXT.1

The TOE provides the following password management capabilities to support strong passwords:

- Passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters:

  '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '~', ',', '[', ']', ':', ';', '|', '_', '/', '.', '<', '>'
- The minimum password length is configurable by the administrator using “user password-size min” command. The password length can be set within the range of 1 to 30 characters.

7.3.3 FIA_UAU.7

There is no TSS requirement for this SFR.

7.3.4 FIA_UIA_EXT.1

The TOE uses password-based authentication to verify users initiating local sessions. The TOE supports public key and password-based authentication for remote SSH users. When a user attempts to establish an SSH session, the TOE verifies that the user has a public key associated and the public key file exists. If these conditions are met, then public key authentication is used, otherwise password authentication is used as the fallback authentication mechanism.

The TOE performs user identification and authentication (both local and remote) using its local database. The TOE maintains authentication data including the user ID, password information, user privileges and roles.

A successful login occurs when the administrator provides a valid username along with either the correct password or verification of the public key. Once identification and authentication succeed, the TOE grants access to the user by showing the CLI prompt.

Before the successful login, the TOE displays an access banner to the user and does not allow any other actions.

7.3.5 FIA_X509_EXT.1/Rev

The TOE performs X.509 certificate validation and certificate path validation according to [RFC5280] during a TLS handshake. The TOE verifies trust on the certificate received from the external audit server and the certificate received from OCSP responder (if applicable).

The TOE performs certificate validation and certificate path validation as follows:

1. Verify that the certificate has a correct format and has not expired.
2. Verify that the chain of trust from the certificate up to the CA root certificate is maintained.
3. Verify that the basicConstraints extension exists and the CA flag is set to TRUE for all CA certificates in the path.
4. Verify that the CA root certificate is trusted.
5. Verify that the certificate has not been revoked.
6. Verify that the extendedKeyUsage field in the certificate corresponds to the use of the certificate (e.g., "Server Authentication", "OCSP Signing").

If all these steps are successful, the certificate is considered valid. If any of these steps fails, the certificate is considered invalid. The TOE does not support certificate pinning.

7.3.6 FIA_X509_EXT.2

The TOE contains a default Certificate Authority (CA) keystore located in the flash filesystem (refer to Table 10). This keystore is used for performing certificate validation and contains all CA root certificates trusted by the TOE.

The TOE obtains the revocation status of the certificate by validating it using the OCSP protocol. If the OCSP responder is not reachable, the validation fails and the certificate is assumed to be revoked.

7.3.7 FIA_X509_EXT.3

Using CLI commands, the administrator can generate a Certificate Signing Request (CSR). After a CSR file is generate, the administrator sends the request to a CA for signing. Once the CA certificate response is received, the administrator uses the CLI commands to validate the certificate chain and import the signed certificate into the TOE. The TOE supports RSA-based as well as ECDSA-based certificates.

7.4 Security Management

7.4.1 FMT_MOF.1/Functions

The TOE transfers audit records to the external audit server using the TLS protocol. The administrator can configure the TLS ciphersuites using the "ssl cipher" command.

The administrator can also configure the handling of audit data as detailed in TSS section for FAU_STG_EXT.1.

The TOE does not allow administrators to modify the behavior when the local audit storage space is full.

7.4.2 FMT_MOF.1/ManualUpdate

There is no TSS requirement for non-distributed TOEs. The TOE is a non-distributed network device.

7.4.3 FMT_MTD.1/CoreData

The TOE provides a Command-Line Interface (CLI) to the administrator for managing the TOE. The TOE does not expose any administrative functions until the administrator has been successfully authenticated.

Access to the trust store and X.509v3 certificates is restricted to administrators via the TOE's filesystem access control policies. Certificates are stored in the /flash/switch/cert.d directory, which acts as a keystore.

7.4.4 FMT_MTD.1/CryptoKeys

The administrator can manage the cryptographic keys as below.

Table 12: Cryptographic Key Management

Cryptographic keys	Usage	Generate	Import	Modify	Delete
SSH host (server) public and private key pair	The TOE authenticates itself to the SSH client	Yes	No	No	Yes
SSH user (client) public key	The TOE authenticates the remote login user over the SSH session	No	Yes	No	Yes
CA root certificate (public key)	The TOE validates X.509 certificates from the external audit server and the code signing certificate in TOE software update images	No	Yes	No	Yes
TOE TLS client certificate (public and private key pair)	The TOE authenticates itself to the external audit server	Yes	Yes	Yes	Yes

7.4.5 FMT_SMF.1

The TOE provides a local administration interface via the console port and a remote administration interface through the SSH protocol. The administrator can perform security management functions through both local and remote interfaces.

7.4.6 FMT_SMR.2

The TOE includes a built-in "admin" account with full privileges to manage the TOE. The "admin" user can create additional configured administrator accounts with varying levels of privileges. The "admin"

user can configure access for other users by modifying read and write permissions on specific modules in the TOE.

Both the built-in and configured administrator accounts are considered Security Administrator in this ST.

Administrators are granted access to management functions based on the access granted to their user account. The TOE can grant read-only or read-write access to the command families which includes file, system, config, module, interface, ip, vlan, dns, qos, policy, session, aaa. The aaa command family, for example, provides the ability to configure the type of authentication methods supported by the TOE and perform user account management.

7.5 Protection of the TSF

7.5.1 FPT_APW_EXT.1

The TOE protects user passwords by hashing them with SHA-256 and storing the resulting hash values in a protected directory on the flash filesystem. User passwords are never stored in plaintext. The TOE does not offer any functions that disclose plaintext passwords to users.

7.5.2 FPT_SKP_EXT.1

The TOE protects pre-shared keys, symmetric keys, and private keys by using the operating system's file access control. The TOE blocks all configured administrators from accessing the files containing the sensitive information mentioned above. The access to these files is restricted to the "root" account (using the “su” command).

7.5.3 FPT_STM_EXT.1

The TOE uses an internal system clock to maintain accurate date and time information. The TOE provides time stamps for the following security functions:

- FAU_GEN.1: The timing of the event is recorded in the audit record.
- FCS_SSH_EXT.1: SSH rekeying is performed when the time limit (maximum minutes) has been reached for the session.
- FIA_AFL.1: The account is locked for a defined period following unsuccessful password-based authentication in remote sessions.
- FIA_X509_EXT.1/Rev: The TOE checks the expiration date of each X.509 certificate in the chain.
- FIA_X509_EXT.2: The TOE checks the validity of the OCSP response and the revocation date included in the OCSP response.
- FIA_X509_EXT.3: The TOE checks the expiration date of each X.509 certificate in the chain.
- FPT_TUD_EXT.1: The TOE checks the expiration date of the code signing certificate.
- FTA_SSL.3: The TOE terminates remote administrative sessions after a predefined idle time interval.
- FTA_SSL_EXT.1: The TOE terminates local administrative sessions after a predefined idle time interval.

7.5.4 FPT_TST_EXT.1

To verify software integrity during initial start-up, the TOE uses the following approach: comparison of calculated SHA2-256 hash values against known-good reference values.

- OS6360, OS6465, OS6465T, OS6560, OS6570, OS6865, and OS9900 platforms: The hash of the complete AOS image file is calculated and compared with the corresponding value in the accompanying hash file (imgsha256sum).

- OS6860N, OS6870 and OS6900 platforms: The hash values of individual images (vmlinuz, initrd) are computed and verified against the reference hashes stored in the boot partition. Those known-good values were previously computed and stored during TOE software installation.

Failure to verify the integrity of the TOE software results in an automatic system reboot.

The TOE integrates the OpenSSL library to implement the underlying cryptographic algorithms. OpenSSL performs power-on self-tests (POST) to ensure the integrity of the module itself and correct operation of cryptographic implementation.

The power-on self-tests include the following:

- Integrity verification over the complete module file image using HMAC-SHA-256;
- Known Answer Test (KAT) for AES encryption and decryption algorithms;
- KAT for DRBG;
- KAT for secure hash algorithms;
- KAT for RSA signature generation and verification algorithms;
- KAT for Diffie-Hellman (DH) algorithm over finite fields;
- KAT for Elliptic Curve Diffie-Hellman (ECDH) algorithm;
- Pair-wise Consistency Tests (PCT) for ECDSA asymmetric algorithms (performing signature generation and verification for a known ECDSA key).

OpenSSL also performs the following conditional self-tests during key generation:

- PCT on each generation of an RSA key pair, consisting of performing signature generation and verification of a predefined message using the generated RSA key pair, as well as public key encryption and private key decryption of a predefined message using the generated RSA key pair.
- PCT on each generation of an ECDSA key pair, consisting of performing signature generation and verification of a predefined message using the generated ECDSA key pair.

In case of failure of any of the power-up self-tests or conditional tests, OpenSSL raises an exception and the TOE shows an error message in the console. As a result, the TOE will reboot itself.

7.5.5 FPT_TUD_EXT.1

The TOE supports manual update of the TOE software. Using CLI commands, the administrator can query and update the TOE software.

To check the TOE software versions, the administrator can use the “show microcode loaded” command for the currently active version, and the “show update history” command to display both the currently active version and the most recently installed version.

Secure installation and update of the TOE software can be performed by an administrator via the CLI as follows:

- Download the new version of TOE software image file from the vendor's website and store the image file in the /flash/working directory.
- Install the downloaded TOE software image using the “reload from working no rollback-timeout” command. If the integrity of the TOE image is verified successfully, the new version of software will be installed, the TOE will reboot upon confirmation from the administrator, and the changes take effect. If the integrity verification fails, the TOE software image is rejected and the command does not proceed with the update.

The TOE does not provide a means for installing a trusted update with a delayed activation.

The TOE software images are signed using a private key securely stored in the development environment, and the corresponding signature and the public-key certificate are embedded in the software images.

After copying the TOE image to the flash filesystem of the TOE, the administrator issues "reload" command to start the installation. The signature verification process is invoked as part of the reload command. First, the signature will be extracted from the image file. Next, the code signing certificate will be validated using the public certificate chain extracted from the image file, up to a CA root certificate in the local trust store. After the code signing certificate is successfully validated, actual signature verification will happen.

The TOE will proceed with rebooting only if the whole verification process is successful. If there is a failure in certificate validation or signature verification, the TOE will not go for reboot and appropriate error message will be displayed.

7.6 TOE Access

In the evaluated configuration, the administrator can access the TOE through Command-Line Interface (CLI) interactive sessions, either locally via the console port or remotely through the SSH client.

7.6.1 FTA_SSL.3

The TOE terminates an idle remote administrative session after an administrator-configurable period of inactivity. The administrator can use the “session cli timeout” command to configure the inactivity timeout value (in minutes). The range of valid values is 1 to 596523.

7.6.2 FTA_SSL.4

The administrator can terminate the administrator’s own session (both local and remote) by issuing the “exit” command.

7.6.3 FTA_SSL_EXT.1

The TOE terminates an idle local administrative session after a period of inactivity. The TOE treats local sessions in the same way as remote sessions. Please refer to TSS section for FTA_SSL.3.

7.6.4 FTA_TAB.1

The TOE can be configured to display an access banner right before and after the administrator logs into the TOE. The same access banner applies to all methods of access: local (through the serial port) and remote (via the SSH session).

Using the “session cli banner” command, the administrator can configure the access banner which appears after user login. Through editing the content of a text file named “pre-banner.txt” in the /flash/switch directory, the administrator can set the access banner displayed before user login.

7.7 Trusted Path/Channels

The TOE implements the TLS protocol version 1.2 (TLS v1.2) and the SSH protocol version 2 (SSHv2) using OpenSSL and OpenSSH, respectively.

7.7.1 FTP_ITC.1

The TOE employs the TLS protocol to provide a trusted communication channel for transferring audit records to a remote syslog server. The TOE acts as the TLS client and authenticates the TLS server using X.509 public-key certificates.

The following table summarizes the trusted communication channel between the TOE and other IT entities:

Table 13: Trusted Channels between TOE and IT Entities

Protocol	Client	Server	Purpose	Authentication of remote IT entity
TLS	TOE	External audit server	Audit data storage	X.509 public-key certificate

7.7.2 FTP_TRP.1/Admin

The TOE provides a trusted path using the SSH protocol for remote administrative sessions. The TOE acts as the SSH server and the administrator executes the SSH client program on the administrator’s workstation.

8 Abbreviations and Terminology

AES
Advanced Encryption System

AOS
Alcatel-Lucent Operating System

API
Application Programming Interface

CA
Certificate Authority

CAVP
Cryptographic Algorithm Validation Program

CBC
Cipher Block Chaining

CC
Common Criteria

CCTL
Common Criteria Testing Laboratory

CLI
Command-Line Interface

CN
Common Name

DH
Diffie-Hellman

DN
Distinguished Name

DNS
Domain Name System

DRBG
Deterministic Random Bit Generator

EAR
Entropy Analysis Report

ECC
Elliptic Curve Cryptography

ECDSA
Elliptic Curve Digital Signature Algorithm

FFC
Finite Field Cryptography

FIPS
Federal Information Processing Standard

GCM

Galois/Counter Mode

HMAC

Hash-based Message Authentication Code

KAS-ECC-SSC

Key Agreement Scheme - Elliptic Curve Cryptography - Shared Secret Computation

KAS-FFC-SSC

Key Agreement Scheme - Finite Field Cryptography - Shared Secret Computation

NIAP

National Information Assurance Partnership

OS

OmniSwitch

PP

Protection Profile

PRF

Pseudorandom Function

RSA

Rivest-Shamir-Adleman

SAN

Subject Alternative Name

SN

Subject Name

SSH

Secure Shell

ST

Security Target

TLS

Transport Layer Security

TOE

Target of Evaluation

TSF

TOE Security Functionality

TSFI

TOE Security Functionality Interface

TSS

TOE Security Summary