Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6560, 6860, 6865, 6900, 9900 with AOS 8.9 R11 Security Target

Version 0.7
October 9, 2023

Prepared for:

ALE USA Inc

2000 Corporate Center Drive

Thousand Oaks, CA 91320

Prepared By:

Gossamer Laboratories

www.gossamersec.com

1. SECURITY TARGET INTRODUCTION 4
1.1 SECURITY TARGET REFERENCE 4
1.2 TOE REFERENCE 4
1.3 TOE OVERVIEW 5
1.3.1 Intended method of use 5
1.3.2 Major security features 6
1.4 TOE DESCRIPTION 6
1.4.1 TOE Architecture 7
1.4.2 TOE Documentation 10
2. CONFORMANCE CLAIMS 11
2.1 CONFORMANCE RATIONALE 12
3. SECURITY OBJECTIVES 13
3.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 13
4. EXTENDED COMPONENTS DEFINITION 15
5. SECURITY REQUIREMENTS 16
5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 16
5.1.1 Security audit (FAU) 17
5.1.2 Cryptographic support (FCS) 20
5.1.3 Identification and authentication (FIA) 23
5.1.4 Security management (FMT) 24
5.1.5 Protection of the TSF (FPT) 25
5.1.6 TOE access (FTA) 26
5.1.7 Trusted path/channels (FTP) 27
5.2 TOE SECURITY ASSURANCE REQUIREMENTS 27
5.2.1 Development (ADV) 27
5.2.2 Guidance documents (AGD) 28
5.2.3 Life-cycle support (ALC) 29
5.2.4 Tests (ATE) 29
5.2.5 Vulnerability assessment (AVA) 30
6. TOE SUMMARY SPECIFICATION 31
6.1 SECURITY AUDIT 31
6.2 CRYPTOGRAPHIC SUPPORT 33
6.2.1 OpenSSL Cryptographic Library 33
6.2.2 Transport Layer Security (TLS) protocol 35
6.2.3 Secure Shell version 2 (SSHv2) protocol 36
6.2.4 Summary 38
6.3 IDENTIFICATION AND AUTHENTICATION 39
6.3.1 Local Authentication 39
6.3.2 X.509 Certificate Generation and Validation 40
6.4 SECURITY MANAGEMENT 41
6.5 PROTECTION OF THE TSF 42
6.6 TOE ACCESS 43
6.7 TRUSTED PATH/CHANNELS 44

LIST OF TABLES

Table 1-1 TOE Hardware Configurations 5
Table 2-1 Technical Decisions 11
Table 5-1 TOE Security Functional Components 17

Table 5-2 Audit events 19
Table 5-3 Assurance Components 27
Table 6-1 TOE audit record levels 31
Table 6-2 Audit local storage 32
Table 6-3 Cryptographic Functions 34
Table 6-4 Key Establishment Schemes 34
Table 6-5 Certificates and keys used by the TLS protocol 36
Table 6-6 SSHv2 algorithms supported by the TOE 37
Table 6-7 Keys used by the SSHv2 protocol 38

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6560, 6860, 6865, 6900, 9900 with AOS 8.9 R11 provided by ALE USA Inc. The TOE is being evaluated as a network device.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

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

1.1 Security Target Reference

ST Title – Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6560, 6860, 6865, 6900, 9900 with AOS 8.9.R11 Security Target

ST Version – Version 0.7

ST Date – October 9, 2023

1.2 TOE Reference

TOE Identification – Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6560, 6860, 6865, 6900, 9900 with AOS 8.9 R11

TOE Developer – ALE USA Inc.

Evaluation Sponsor – ALE USA Inc.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Alcatel-Lucent Enterprise OmniSwitch series 6360, 6465, 6560, 6860, 6865, 6900, and 9900 with AOS 8.9 R11. The firmware is named Alcatel-Lucent Operating System (AOS) which is the single purpose operating system that operates the management functions of all of the Alcatel-Lucent Enterprise OmniSwitch switches. The evaluation covers AOS 8. 9 R11.

The TOE hardware consists of the following OminSwitch models:

Model	Processor ID	Microarchitecture	Network Interface
OmniSwitch 6360 (OS6360)	Marvell 98DX236S	ARM Cortex-A9	Marvell AlleyCat 3
	Marvell 98DX233S	ARM Cortex-A9	Marvell AlleyCat 3
OmniSwitch 6465 (OS6465/OS6465T)	Marvell 98DX3233	ARM Cortex-A9	Marvell AlleyCat 3
OmniSwitch 6560 (OS6560)	Marvell 88F6820	ARM Cortex-A9	Marvell AlleyCat 3
OmniSwitch 6860 (OS6860E)	Broadcom BCM56342	ARM Cortex-A9	Broadcom Helix4
	Broadcom BCM56340	ARM Cortex-A9	Broadcom Helix4
OmniSwitch 6860 (OS6860N)	Intel Atom C3338	Goldmont	Broadcom Trident3
	Intel Atom C3558	Goldmont	Broadcom Trident3
OmniSwitch 6865 (OS6865)	Broadcom BCM56342	ARM Cortex-A9	Broadcom Helix4
OmniSwitch 6900 (OS6900)	Intel Atom C3558	Goldmont	Broadcom Trident3
	Intel Atom C2538	Rangeley	Broadcom Tomahawk
	Intel Xeon D1518	Broadwell	Broadcom Trident3
OmniSwitch 9900 (OS9900)	Intel Atom C2518	Rangeley	Marvell Prestera DX

Table 1-1 TOE Hardware Configurations

The TOE provides Layer-2 switching, Layer-3 routing, and traffic filtering. Layer-2 switching analyzes incoming frames and makes forwarding decisions based on information contained in the frames. Layer-3 routing determines the next network point to which a packet should be forwarded toward its destination. These devices may create or maintain a table of the available routes and their conditions and use this information along with distance and cost algorithms to determine the best route for a given packet. Routing protocols include Border Gateway Protocol (BGP), Routing Information Protocol (RIP) v.2, and Open Shortest Path First (OSPF). Filtering controls network traffic by controlling whether packets are forwarded or blocked at the TOE's interfaces. Each packet is examined to determine whether to forward or drop the packet, on the basis of the criteria specified within the access lists. Access list criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information.

All series perform the same security functions with respect to this evaluation. The differences between the models are in speed and physical characteristics.

1.3.1 Intended method of use

The intended TOE environment is a secure data center that protects the TOE from unauthorized physical access. Only security administrators are to have access to connect to the serial console, or gain physical access to the hardware. Appropriate administrator security policy and security procedure guidance must be in place to govern operational management of the TOE within its operational environment.

The TOE is not intended for use as a general purpose computer and only executes the services needed to perform its intended function.

1.3.2 Major security features

The TOE provides the following security functions.

- Generation of audit records for security related events, which can be locally stored or sent to a remote server
- Cryptographic support for protecting TOE Security Functionality (TSF) data, password storage, trusted update of the TOE firmware, self-tests, and for establishing secure protocols used by the TOE
- Identification and authentication of Security Administrators that access the TOE for Security Management purposes
- Security management of the TSF, via a Command Line Interface (CLI) using local and remote sessions.
- Protection of the TSF, through the establishment of secure channels between the TOE and external IT entities, remote consoles or other devices in the network; protection of passwords stored in the TOE; updates of the TOE firmware using trusted product updates; and provision of reliable timestamps

1.4 TOE Description

The following diagram shows the basic components that comprise the TOE.

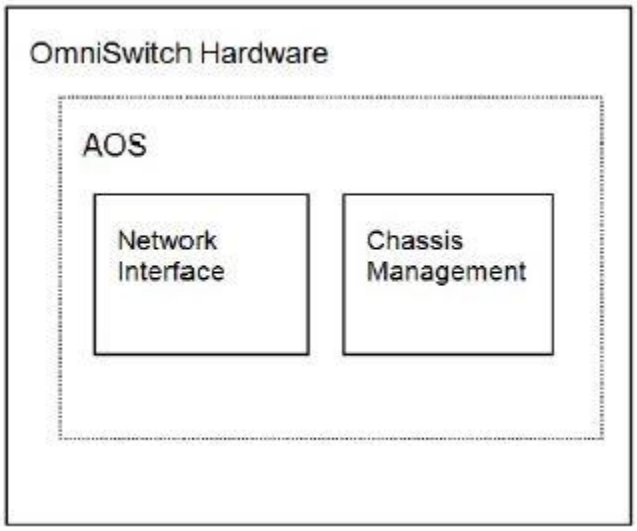

Figure 1 TOE Architecture

The term Chassis Management Module (CMM) is used to describe the logical management functionality of the TOE providing the following services.

- Console, Universal Serial Bus (USB), and Ethernet management port connections. The console port that is used to connect a serial console to initialize and configure the TOE via a Command Line Interface (CLI). Depending on the TOE model the physical interface can be an USB or an RJ-45 connector
- Software and configuration management, including the CLI
- Power distribution
- Diagnostics
- Cryptographic functionality
- Important availability features, including failover (when used in conjunction with another CMM), software rollback, temperature management, and power management

Network Interface (NI) modules provides the connectivity to the network through different physical ports, connector types and speed. The NI modules are categorized into Gigabit Ethernet Network Interface (GNI), 10-Gigabit Ethernet Network Interface (XNI), and 40-Gigabit Ethernet Network Interface (QNI) modules. GNI modules provide 1000 Mbps (1 Gbps) connections. GNI modules can be used for backbone connections in networks where Gigabit Ethernet is used as the backbone media. XNI modules provide up to six 10000 Mbps (10 Gbps) connections per module and can be used in networks where 10-gigabit Ethernet is used as the backbone media. Finally, QNI modules provide 40000 Mbps (40 Gbps) connections per module.

The main distinction between models are the form factor (either chassis or stacks), the processor used, the number of physical ports, the port speeds, the connector types, and the amount of physical RAM installed.

The OS6360, OS6465, OS6465T, OS6560, OS6860E, OS6860N, OS6865 series products are packaged in a single PCB with an embedded CPU Cortex ARM 9 processor or Intel Atom processor as shown in Table 1-1. The CMM and NI functions execute on this processor, and communicate via a socket based protocol running over TCP/IP.

The OS6900 series products are packaged in a single PCB with an Intel processors shown in Table 1-1. The CMM and NI functions execute on this processor, and communicate via a socket-based protocol running over TCP/IP.

The OS9900 is a chassis-based product including a CMM with an Intel Atom C2518processor. The CMM functions execute on this processor and communicate with the NIs via a socket-based protocol running over TCP/IP. This product can support up to six NI cards containing an Intel Atom C2518 processor, where the NI functions execute. The cryptographic functionality that the TOE uses in the evaluated configuration does not run on the NI cards.

An OmniSwitch 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 bridge and router. Virtual chassis connects two or more physical stackable switches through Virtual Fabric Links (VFL) and has a specific protocol to communicate between switches.

Virtual Chassis mode is not allowed in the evaluated configuration. The TOE must always operate in Standalone mode.

1.4.1 TOE Architecture

Each TOE appliance runs the AOS 8.9 R11 software and has physical network connections to its environment to facilitate managing and filtering network traffic. Figure 1 shows the TOE's physical boundary.

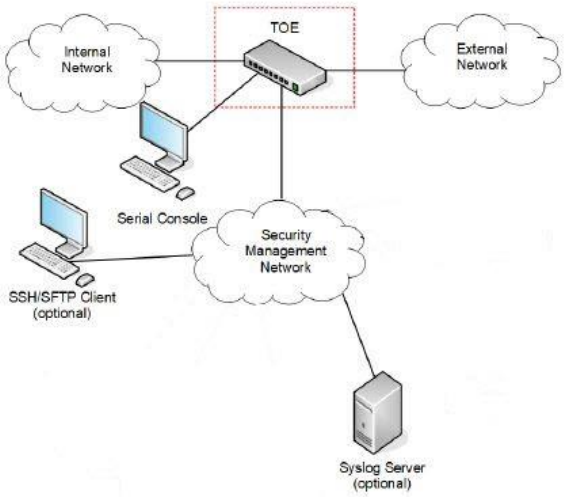

Figure 2 TOE Boundary

1.4.1.1 Physical Boundaries

Figure 2 shows a depiction of the TOE and its operating environment. The red dotted line encloses the TOE physical boundary.

- The TOE is located between the external and the internal network or within the internal network of an organization in order to perform Layer-2 switching, Layer-3 routing, and traffic filtering of flowing IP packets. The TOE can be also connected to an external IT entity (e.g., Syslog Server).
- Administrators log onto 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 an SSHv2 client from a computer connected to the security management network.
- Administrators can execute commands from the CLI to connect to external SSH via the SSHv2 protocol.
- TOE audit records can be optionally stored in a Syslog Server. Communication is protected by the TLS protocol.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the ALE OmniSwitch with AOS 8.9 R11:

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

1.4.1.2.1 Security audit

The TOE generates audit records. The audit records can be displayed on the serial console as they are generated in a scrolling format.

The TOE writes audit records to a set of circular files stored in the systems 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 also provides the ability to send the audit records to an external syslog server using a secure channel.

The TOE provides to security administrators the ability to modify the maximum size allowed for the audit files. Once the files are full the oldest entries are overwritten.

1.4.1.2.2 Cryptographic support

The TOE requires cryptography for supporting the following functionality.

- Establishment of secure channels using the SSHv2 and TLSv1.2 protocols
- X.509 certificate generation and validation
- Storage of passwords
- Self-tests of the cryptographic algorithms
- Verification of the integrity of the TOE firmware

The TOE provides cryptographic support using the OpenSSL and OpenSSH software packages, which are bundled in the TOE.

1.4.1.2.3 Identification and authentication

The TOE requires identification and authentication of administrators of the TOE prior to access any of the management functionality in all possible scenarios, which are as follows.

- TOE administrators accessing (either locally or remotely) the Command Line Interface (CLI) via a serial console or a Secure Shell (SSH) session

The TOE displays to the administrator a configurable banner before the administrator successfully logs onto the TOE (either serial console or SSH). The TOE also provides the ability to lock the administrator after a configurable number of unsuccessful attempts, and terminate the logon session after a configurable period of inactivity.

The TOE provides administrator configurable password settings to enforce password complexity when a password is created or modified.

The TOE provides support for the following Identification and Authentication mechanisms.

- Identification and Authentication made by the TOE using credentials stored in the local file system
- Communication with SSH clients is protected with the Secure Shell (SSH) protocol.

1.4.1.2.4 Security management

The TOE provides a Command-Line Interface (CLI) for security management. TOE administrators connect to the TOE via either a serial console or a remote session using Secure Shell (SSHv2). In either case, administrators are required to identify and authenticate against the TOE before getting access to the CLI.

1.4.1.2.5 Protection of the TSF

The TOE protects itself by requiring administrators to identify and authenticate themselves prior to performing any actions and by defining the access allowed by each administrator. The TOE uses the filesystem access control to protect access to sensitive data like cryptographic keys and credentials.

The TOE ensures that manual updates of the TOE firmware are done using trusted updates by verifying the integrity of the new version of the TOE firmware.

The TOE also implements self-tests to ensure the correct operation of cryptographic services.

The TOE also provides a reliable date and time that is used for audit record timestamps, certificate verification and session timing.

1.4.1.2.6 TOE access

The TOE can be configured to display a login banner when an administrator establishes an interactive session and subsequently will enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated.

1.4.1.2.7 Trusted path/channels

The TOE provides the following secure channels to ensure the integrity and confidentiality of the information exchanged between the TOE and external IT entities in the operational environment.

- Transport Layer Security (TLS) versions 1.2 is used to protect communication with external audit servers (syslog).
- Secure Shell version 2 (SSHv2) is used to protect communication with SSH clients.

1.4.2 TOE Documentation

ALE offers a series of documents that describe the installation of the TOE as well as guidance for subsequent use and administration of the applicable security features of the OmniSwitch with AOS 8.9 R11. The following document was examined as part of the evaluation:

- Preparation and Operation of Common Criteria Evaluated OmniSwitch Products (NDcPP), AOS Release 8.9 R11, July 2023.

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - 'collaborative Protection Profile for Network Devices', Version 2.2e, 23 March 2020 (NDcPP22e)

Table 2-1 Technical Decisions

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

CPP_ND_V2.2E	TD0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0570 - NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
CPP_ND_V2.2E	TD0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	Yes	
CPP_ND_V2.2E	TD0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
CPP_ND_V2.2E	TD0563 - NiT Technical Decision for Clarification of audit date information	Yes	
CPP_ND_V2.2E	TD0556 - NIT Technical Decision for RFC 5077 question	Yes	
CPP_ND_V2.2E	TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes	
CPP_ND_V2.2E	TD0547 - NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
CPP_ND_V2.2E	TD0546 - NIT Technical Decision for DTLS - clarification of Application Note 63	No	Requirement not claimed
CPP_ND_V2.2E	TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
CPP_ND_V2.2E	TD0536 - NIT Technical Decision for Update Verification Inconsistency	Yes	
CPP_ND_V2.2E	TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	Requirement not claimed
CPP_ND_V2.2E	TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	

2.1 Conformance Rationale

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

3. Security Objectives

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

In general, the NDcPP22e has defined Security Objectives appropriate for network devices and as such are applicable to the ALE OmniSwitch with AOS 8.9 R11 TOE.

3.1 Security Objectives for the Operational Environment

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

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

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

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

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

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

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

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

OE.VM_CONFIGURATION (applies to vNDs only)
For vNDs, the Security Administrator ensures that the VS and VMs are configured to

- reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and
- correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

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

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

4. Extended Components Definition

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

Extended SFRs:

- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP22e:FAU_STG_EXT.3/LocSpace: Action in case of possible audit data loss
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
- NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0634
- NDcPP22e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
- NDcPP22e:FIA_PMG_EXT.1: Password Management
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
- NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
- NDcPP22e:FPT_TST_EXT.1: TSF testing
- NDcPP22e:FPT_TUD_EXT.1: Trusted update
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking

5. Security Requirements

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

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

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

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by ALE OmniSwitch with AOS 8.9 R11 TOE.

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

	NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP22e:FMT_SMF.1: Specification of Management Functions - per TD0631
	NDcPP22e:FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
	NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
	NDcPP22e:FPT_TST_EXT.1: TSF testing
	NDcPP22e:FPT_TUD_EXT.1: Trusted update
FTA: TOE access	NDcPP22e:FTA_SSL.3: TSF-initiated Termination
	NDcPP22e:FTA_SSL.4: User-initiated Termination
	NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP22e:FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel - per TD0639
	NDcPP22e:FTP_TRP.1/Admin: Trusted Path - per TD0639

Table 5-1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1)

NDcPP22e:FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [***[Starting and stopping services]***];

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

Requirement	Auditable Events	Additional Content
NDcPP22e:FAU_GEN.1	None	None
NDcPP22e:FAU_GEN.2	None	None
NDcPP22e:FAU_STG.1	None	None
NDcPP22e:FAU_STG_EXT.1	None	None
NDcPP22e:FAU_STG_EXT.3/LocSpace	Low storage space for audit events.	None
NDcPP22e:FCS_CKM.1	None	None
NDcPP22e:FCS_CKM.2	None	None
NDcPP22e:FCS_CKM.4	None	None
NDcPP22e:FCS_COP.1/DataEncryption	None	None
NDcPP22e:FCS_COP.1/Hash	None	None

NDcPP22e:FCS_COP.1/KeyedHash	None	None
NDcPP22e:FCS_COP.1/SigGen	None	None
NDcPP22e:FCS_RBG_EXT.1	None	None
NDcPP22e:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP22e:FCS_TLSC_EXT.1	Failure to establish a TLS Session.	Reason for failure.
NDcPP22e:FCS_TLSC_EXT.2	None	None
NDcPP22e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_PMG_EXT.1	None	None
NDcPP22e:FIA_UAU.7	None	None
NDcPP22e:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
NDcPP22e:FIA_X509_EXT.2	None	None
NDcPP22e:FIA_X509_EXT.3	None	None
NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP22e:FMT_MTD.1/CoreData	None	None
NDcPP22e:FMT_MTD.1/CryptoKeys	None	None
NDcPP22e:FMT_SMF.1	All management activities of TSF data.	None
NDcPP22e:FMT_SMR.2	None	None
NDcPP22e:FPT_APW_EXT.1	None	None
NDcPP22e:FPT_SKP_EXT.1	None	None
NDcPP22e:FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
NDcPP22e:FPT_TST_EXT.1	None	None
NDcPP22e:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
NDcPP22e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
NDcPP22e:FTA_SSL.4	The termination of an interactive session.	None
NDcPP22e:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	None

NDcPP22e:FTA_TAB.1	None	None
NDcPP22e:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
NDcPP22e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None

Table 5-2 Audit events

NDcPP22e:FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of **Table 5-2 Audit events.**

5.1.1.2 User identity association (NDcPP22e:FAU_GEN.2)

NDcPP22e:FAU_GEN.2.1

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

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

NDcPP22e:FAU_STG.1.1

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

NDcPP22e:FAU_STG.1.2

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

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

NDcPP22e:FAU_STG_EXT.1.1

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

NDcPP22e:FAU_STG_EXT.1.2

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

NDcPP22e:FAU_STG_EXT.1.3

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

5.1.1.5 Action in case of possible audit data loss (NDcPP22e:FAU_STG_EXT.3/LocSpace)

NDcPP22e:FAU_STG_EXT.3.1/LocSpace

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

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

NDcPP22e:FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

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

5.1.2.2 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

NDcPP22e:FCS_CKM.2.1

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

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

5.1.2.3 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

NDcPP22e:FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***instructs a part of the TSF to destroy the abstraction that represents the key***]

that meets the following: No Standard.

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

NDcPP22e:FCS_COP.1.1/DataEncryption

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

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

NDcPP22e:FCS_COP.1.1/Hash

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

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

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

NDcPP22e:FCS_COP.1.1/KeyedHash

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

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

NDcPP22e:FCS_COP.1.1/SigGen

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

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 and 3072 bits],***
- ***Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]***]

that meet the following: [

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

5.1.2.8 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

NDcPP22e:FCS_RBG_EXT.1.1

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

NDcPP22e:FCS_RBG_EXT.1.2

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

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

NDcPP22e:FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol that complies with: RFC(s) 4251, 4252, 4253, 4254, [***4344, 5656, 6668, 8332***].

NDcPP22e:FCS_SSHS_EXT.1.2

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

NDcPP22e:FCS_SSHS_EXT.1.3

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

NDcPP22e:FCS_SSHS_EXT.1.4

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

NDcPP22e:FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [***rsa-***

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

NDcPP22e:FCS_SSHS_EXT.1.6

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

NDcPP22e:FCS_SSHS_EXT.1.7

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

NDcPP22e:FCS_SSHS_EXT.1.8

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

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

NDcPP22e:FCS_TLSC_EXT.1.1

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

TLS_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 4492,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
TLS_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.

NDcPP22e:FCS_TLSC_EXT.1.2

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

NDcPP22e:FCS_TLSC_EXT.1.3

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

NDcPP22e:FCS_TLSC_EXT.1.4

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

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

NDcPP22e:FCS_TLSC_EXT.2.1

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

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

NDcPP22e:FIA_AFL.1.1

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

NDcPP22e:FIA_AFL.1.2

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

5.1.3.2 Password Management (NDcPP22e:FIA_PMG_EXT.1)

NDcPP22e:FIA_PMG_EXT.1.1

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

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

b) Minimum password length shall be configurable to between [**1**] and [**30**] characters.

5.1.3.3 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

NDcPP22e:FIA_UAU.7.1

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

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

NDcPP22e:FIA_UAU_EXT.2.1

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

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

NDcPP22e:FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- [***no other actions***].

NDcPP22e:FIA_UIA_EXT.1.2

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

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

NDcPP22e:FIA_X509_EXT.1.1/Rev

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

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

NDcPP22e:FIA_X509_EXT.1.2/Rev

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

5.1.3.7 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

NDcPP22e: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***].

NDcPP22e:FIA_X509_EXT.2.2

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

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

NDcPP22e:FIA_X509_EXT.3.1

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

NDcPP22e:FIA_X509_EXT.3.2

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

5.1.4 Security management (FMT)

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

NDcPP22e:FMT_MOF.1.1/ManualUpdate

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

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

NDcPP22e:FMT_MTD.1.1/CoreData

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

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

NDcPP22e:FMT_MTD.1.1/CryptoKeys

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

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

NDcPP22e:FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***hash comparison***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),***
- ***Ability to manage the cryptographic keys,***
- ***Ability to configure the cryptographic functionality,***
- ***Ability to 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 import X509v3 certificates to the TOE's trust store***].

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

NDcPP22e:FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

NDcPP22e:FMT_SMR.2.2

The TSF shall be able to associate users with roles.

NDcPP22e:FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

5.1.5 Protection of the TSF (FPT)

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

NDcPP22e:FPT_APW_EXT.1.1

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

NDcPP22e:FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

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

NDcPP22e:FPT_SKP_EXT.1.1

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

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

NDcPP22e:FPT_STM_EXT.1.1

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

NDcPP22e:FPT_STM_EXT.1.2
The TSF shall [***allow the Security Administrator to set the time***].

5.1.5.4 TSF testing (NDcPP22e:FPT_TST_EXT.1)

NDcPP22e:FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [
- **power on self-tests required by the FIPS 140-3 standard in the OpenSSL Cryptographic Library;**
- **verification of TOE executable using published hash**].

5.1.5.5 Trusted update (NDcPP22e:FPT_TUD_EXT.1)

NDcPP22e: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***].

NDcPP22e:FPT_TUD_EXT.1.2
The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].

NDcPP22e:FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***published hash***] prior to installing those updates.

5.1.6 TOE access (FTA)

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

NDcPP22e:FTA_SSL.3.1
The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

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

NDcPP22e:FTA_SSL.4.1
The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

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

NDcPP22e:FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

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

NDcPP22e:FTA_TAB.1.1
Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.7 Trusted path/channels (FTP)

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

NDcPP22e: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.

NDcPP22e:FTP_ITC.1.2
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

NDcPP22e:FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [**sending audit records to the external syslog server**].

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

NDcPP22e:FTP_TRP.1.1/Admin
The TSF shall be capable of using [***SSH, TLS***] to provide a communication path between itself and authorized remote Administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

NDcPP22e:FTP_TRP.1.2/Admin
The TSF shall permit remote Administrators to initiate communication via the trusted path.

NDcPP22e:FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

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

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

Table 5-3 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d
The developer shall provide a functional specification.

ADV_FSP.1.2d
The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d
The developer shall provide operational user guidance.

AGD_OPE.1.1c
The operational user guidance shall describe, for each user role, the user accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c
The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c
The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c
The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c
The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6c
The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c
The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d
The developer shall provide the TOE, including its preparative procedures.

AGD_PRE.1.1c
The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c
The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e
The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e

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

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

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

AVA_VAN.1.3e

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

6. TOE Summary Specification

This chapter describes the security functions:

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

6.1 Security audit

Audit functionality in the TOE is provided via the switch logging feature, which records audit events for all administrative operations performed. 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 certificate management tasks (e.g., generation, import, update, deletion, view and verification of certificates), 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.

The TOE supports the severity levels detailed in

Security Level	Type	Description
1 (highest severity)	Alarm	A serious, non-recoverable error has occurred and the system must be rebooted.
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.

Table 6-1 TOE audit record levels

Security level 6 (Info) is enabled for all events by default and is the minimum severity level required in the evaluated configuration.

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 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 security level 6 (Info) and their description prefixed with the "EVENT-AUDIT". It is possible to configure the severity level either globally for all applications or on a per application basis.

The TOE can be configured to send records to an audit file on the flash file system, display them on the serial console, and/or send them to a remote syslog server.

For local permanent storage, audit data is stored in the audit file set located in the flash file system and prefixed as "swlog_". Whenever the audit file reaches a configurable threshold (maximum size for audit files), the audit file overwrites the content of the oldest audit file in the set of circular audit files (in case the set is full). The amount of audit data that can be generated depends on the maximum size allowed per audit file, which can be changed using the command: ***swlog output flash file-size <size in KB>.*** The TOE also provides a mechanism to display a warning to the administrator if the storage capacity has reached 90% of the configured size in any of the audit files.

Audit files are protected from modification and deletion by enforcing filesystem access control on groups of commands. Only the Security Administrator has privileges to clear the audit records.

The following table shows the number of circular files that comprise the audit file set, the file names for the audit file set, the parameter used to define the maximum size allowed for audit records, and the default value and allowed range for that parameter.

Number of Circular Files	Audit File Set	Parameter Definition	Default Value	Allowed Values
Eight	swlog_<suffix>, swlog_<suffix>.0 through swlog_<suffix>.6	Maximum size for each file (in KB)	1250KB	125KB to 12500KB

Table 6-2 Audit local storage

The TOE can also transfer audit data to an external audit server, implemented with a syslog server. A secure communication channel is established between the TOE and the external audit server using TLSv1.2 in order to protect audit data communication from loss of integrity or confidentiality.

An audit record is sent to the audit server immediately after the event occurs. If communication fails, the audit event is only recorded locally and is not resent; the TOE tries to reconnect to the audit server whenever a new audit generation request is received

The Security audit function satisfies the following security functional requirements:

- NDcPP22e:FAU_GEN.1 The TOE generates all the required audit events in Table 5-2. Each audit record identifies the date/time, event type, outcome of the event, responsible subject/user, as well as the additional event-specific content indicated in Table 5-2.
- NDcPP22e:FAU_GEN.2: The TOE identifies the responsible user for each event based on the specific administrator or network entity (identified by IP address) that caused the event.
- NDcPP22e:FAU_STG.1: The audit files are protected from deletion and modification. Only the Security Administrator can clear the audit records.
- NDcPP22e:FAU_STG_EXT.1: Audit events are stored locally in the flash memory using a circular chain of audit files. Whenever the audit file reaches a configurable threshold (maximum size for audit files), the TOE overwrites the content of the oldest audit file in the set of circular audit files (in case the set is full).

  The TOE audit functionality can be also configured to send the audit events to an external syslog server using TLS for protecting the communication channel.
- NDcPP22e:FAU_STG_EXT.3/LocSpace: The TOE generate a warning to inform the Administrator if the audit trail exceeds the local audit trail storage capacity.

6.2 Cryptographic support

The TOE implements cryptographic protocols and algorithms using the following standard packages included in the TOE.

- OpenSSL v3.0.7 for TLSv1.2, and cryptographic algorithm support
- OpenSSH v8.8p1 for implementing the SSHv2 protocol. OpenSSH uses OpenSSL for the underlying cryptographic algorithms

6.2.1 OpenSSL Cryptographic Library

The TOE includes OpenSSL, which implements version 1.2 of the Transport Layer Security (TLS) protocol and provides general-purpose cryptographic services. The following table summarizes the cryptographic algorithms implemented in OpenSSL and used by the TOE to support communication protocols, protection of TSF data and authentication.

Functions	Requirement	Standard	Certificate #
Encryption/Decryption			
AES CBC (128 and 256 bits)	FCS_COP.1/DataEncryption	FIPS Pub 197 ISO 10116 NIST SP 800-38A ISO 19772	A3688
AES-CTR (128 and 256 bits)	FCS_COP.1/DataEncryption	FIPS Pub 197 ISO 10116 NIST SP 800-38A ISO 19772	A3688
AES GCM (128 and 256 bits)	FCS_COP.1/DataEncryption	ISO 19772 FIPS Pub 197 NIST SP 800-38A	A3688
Cryptographic hashing			
SHA-1, SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	FIPS Pub 180-4 ISO/IEC 10118-3:2004	A3688
Keyed-hash message authentication			
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 (digest sizes and block sizes of 256, 384 and 512 bits)	FCS_COP.1/KeyedHash	FIPS Pub 198-1 FIPS Pub 180-4 ISO/IEC 9797-2:2011	A3688
Cryptographic signature services			
RSA Digital Signature (rDSA) (2048, 3072 bits)	FCS_COP.1/SigGen	FIPS Pub 186-4 ISO/IEC 9796-2	A3688
ECDSA Digital Signature (P-256, P-384, P-521)	FCS_COP.1/SigGen	FIPS Pub 186-4 ISO/IEC 14888-3	A3688
Random bit generation			
CTR_DRBG(AES), Hash_DRBG, HMAC_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism	FCS_RBG_EXT.1	FIPS SP 800-90A ISO/IEC 18031:2011	A3688
Key generation			
RSA Key Generation (2048-bit)	FCS_CKM.1	FIPS Pub 186-4 ISO/IEC 9796-2	A3688
ECC Key Generation (P-256, P-384, P-521)	FCS_CKM.1	FIPS PUB 186-4	A3688
FFC Scheme using Diffie-Hellman Group 14	FCS_CKM.1	NIST SP 800-56A Rev 3	Tested with known good impl
Key establishment			

RSA	FCS_CKM.2	RSAES-PKCS1-v1_5	Tested with known good impl
KAS ECC	FCS_CKM.2	NIST SP 800-56A Rev 3	A3688
FFC Schemes using 'safe-prime' groups	FCS_CKM.2	NIST SP 800-56A Rev 3	Tested with known good impl

Table 6-3 Cryptographic Functions

Scheme	SFRs	Service
RSA key establishment	FCS_SSHS_EXT.1	Remote Administration (SSHv2)
ECC key establishment	FCS_SSHS_EXT.1	Remote Administration (SSHv2)
FFC Safe-primes key establishment	FCS_SSHS_EXT.1 (DH 14)	Remote Administration (SSHv2)
RSA key establishment	FCS_TLSC_EXT.1	Remote Audit Server (syslog)
ECC key establishment	FCS_TLSC_EXT.1	Remote Audit Server (syslog)

Table 6-4 Key Establishment Schemes

The TOE OpenSSL library includes a Deterministic Random Bit Generator (DRBG) used for key generation and random data (e.g. shared secrets). The DRBG uses the Hash_DRBG with SHA-256 algorithm by default; if there is a failure in the instantiation, the DRBG will fall back to CTR_DRBG using the AES-256 algorithm, and then to HMAC_DRBG using the HMAC-SHA-256 algorithm.

The TOE OpenSSL library uses a Non-Deterministic Random Number Generator (NDRNG) as the entropy source for seeding the DRBG. The NDRNG is provided by the operating system kernel and based on HAVEGE (Hardware Volatile Entropy Gathering and Expansion) that uses the uncertainties which appear in the behavior of the processor when interrupts occur. The NDRNG provides at least 256 bits of entropy.

OpenSSL performs the following power-up self-tests to ensure that the module and all validated cryptographic algorithms work properly.

- Integrity verification of the shared libraries that comprise the Cryptographic Library
- Known Answer Test (KAT) for symmetric encryption and decryption algorithms
- KAT for the DRBG
- KAT for MAC and message digest algorithms
- KAT for RSA signature generation and verification algorithms
- KAT for the Elliptic Curve Diffie-Hellman algorithm
- Pair-wise Consistency Tests (PCT) for ECDSA asymmetric algorithms (consisting of performing signature generation and verification for a known ECDSA key.

OpenSSL also performs the following conditional tests during the execution of services:

- 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.

OpenSSL maintains all secret keys, private keys, public keys, certificates and other critical security parameters (CSP) used by the cryptographic services (DRBG internal state, session keys, etc.) requested by the TOE in random access memory (RAM) during the life-time of the cryptographic operation. All CSPs are in RAM in plaintext form; OpenSSL clears with zeroes and deallocates all the memory used by the CSP when they are no longer needed (e.g. the cryptographic handler is freed or a TLS session is finished).

6.2.2 Transport Layer Security (TLS) protocol

The TOE implements version 1.2 of the TLS protocol provided by OpenSSL. The TOE establishes a secure channel using TLS for the following purposes.

- As a TLS client
  - Communication with an external audit server (syslog) for audit storage

The TOE implements mutual authentication when acting as a TLS client through the use of X.509 certificates. The TOE performs certificate and 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. See section 6.3.4 for more information.

When acting as a client, for single authentication, it is the server end who presents the certificate during TLS handshake, so when acting as a client, the TOE only parses the certificate from the TLS message and verifies that the server certificate is valid. For mutual authentication, though, the TOE also has to send the client certificate at the server's request. The TOE looks for the certificate and its private key at the certificate directory (/flash/switch/cert.d). Table 6-5 lists the naming conventions for certificates used for communication with external entities.

When acting as a TLS client, the TOE verifies that the certificate presented by the TLS server during the TLS handshake corresponds to the server, using one of the following methods.

- If the server certificate includes a DNS reference identifier in a Subject Alternative Name (SAN) field, verify that the DNS reference identifier matches the server hostname, according to [RFC6125]/
- If the server certificate does not include a DNS reference identifier in a Subject Alternative Name (SAN) field, verify that the DNS reference identifier provided in the Common Name (CN) matches the server hostname, according to [RFC6125].

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

The TOE does not support the use of wildcards in DNS names included in certificates.

The TOE only allows the establishment of a TLS secure channel using TLSv1.2.

The TOE creates session keys following the TLS protocol specification and using the DRBG implemented in OpenSSL. The TOE destroys session keys when the session is terminated by clearing with zeroes and deallocating the RAM memory used to store the session keys.

The TOE supports the following cipher suites.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

The cipher suites are selected in the order shown in the list. In addition, the administrator can further restrict the cipher suites that the TOE shall use via management functions (e.g. ssl cipher CLI command).

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 disable it.

Key agreement parameters used for server key exchange by ECDHE cipher suites are determined based on the chosen elliptic curve, negotiated between the TLS client and server from the list of NIST curves aforementioned.

For destruction of plaintext keys in volatile storage, the TOE relies on the functionality provided by OpenSSL to zeroize and release the memory allocated for cryptographic operations. For keys in non-volatile storage, the TOE provides CLI commands to remove the relationship between keys and certificates and their usage in the TOE.

The following table shows the cryptographic keys involved in the TLS protocol, their location and how they are created and destroyed.

Certificate/Key	Certificate/Key Location	Purpose	Destruction
TOE client certificate (public and private keys)	/flash/switch/cert.d /myCliCert.pem (or .crt) /flash/switch/cert.d /myCliPrivate.key	TOE authentication (TOE acting as client)	Removal from filesystem through CLI commands
External Entity server certificate (publickey)	RAM (received from TLS server)	External Entity authentication (TOE acting as client) Key establishment	Zeroization and deallocation when session is terminated
External Entity client certificate (publickey)	RAM (received from TLS client)	External entity authentication (TOE acting as server)	Zeroization and deallocation when session is terminated
Session keys (shared secrets, ephemeral keys, encryption keys, data authentication keys)	RAM	Integrity and confidentiality during session	Zeroization and deallocation when session is terminated

Table 6-5 Certificates and keys used by the TLS protocol

6.2.3 Secure Shell version 2 (SSHv2) protocol

The TOE implements the Secure Shell version 2 (SSHv2) protocol using OpenSSH v8.8p1. The package uses OpenSSL as the underlying layer for cryptographic algorithms.

The TOE establishes a secure channel using SSHv2 for the following purposes.

- Secure communication for SSHv2 clients used in Security Management (Command Line Interface)

The SSHv2 protocol complies with [RFC4251], [RFC4252], [RFC4253], [RFC4254], [RFC4344], [RFC5656], [RFC6668] and [RFC8332]. The TOE also implements Diffie-Hellman group 14 in accordance with [RFC3526] section 3.

The following table shows the algorithms used for the different aspects of the SSHv2 protocol in the AOS supported by the TOE.

SSHv2 protocol aspect	Cryptographic Algorithm
User Authentication Methods	Public-key based using RSAPKCS#1v1.5 and ECDSA (see public key algorithms)
	Password based using SHA-256.
Key Establishment (key exchange)	Diffie Hellman group 14with SHA-1, group 14 with SHA-256 and group 16 w/ SHA-512
	Elliptic Curve Diffie Hellman with SHA-256, SHA-384 and SHA-512, and NIST curves P-256, P-384 and P-521
Encryption algorithms	AES (CBC mode) with 128-bit and 256-bit keys
	AES (CTR mode) with 128-bit and 256-bit keys
	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 SHA-256 and SHA-512 (rsa-sha2-256 or rsa-sha2-512), using 2048-bit and 3072-bit keys.
	ECDSA withSHA-256, SHA-384 and SHA-512, and NIST curves P-256, P-384 and P-521
Data integrity MAC algorithms	HMAC-SHA1, HMAC-SHA1-96
	HMAC-SHA-256
	HMAC-SHA-512

Table 6-6 SSHv2 algorithms supported by the TOE

The TOE implements zlib compression according to the "OPTIONAL" method specified in section 6.2 of [RFC4253]. Apart from this optional characteristic and with the exception of the cryptographic algorithms mentioned in Table 18 that are marked as "OPTIONAL" in the aforementioned RFCs, the TOE does not implement additional optional features.

The TOE (acting as a SSHv2 server) supports SSHv2 sessions using SSH public key and password authentication by default. When a user attempts to establish a SSHv2 session, the TOE verifies that the user has a public key associated and the public key file exists (/flash/switch/.profiles/.<username>_keys.pub). If these conditions are met, then public key authentication is used, otherwise password authentication is used as the fallback authentication mechanism.

In addition, the TOE provides the ***ssh enforce pubkey-auth*** command in the CLI to enforce public key authentication only, thus disabling password-based authentication. If public key authentication fails, then access to the TOE is not granted.

The TOE limits the size of SSHv2 packets to 262126 bytes; packets greater than this size in an SSH transport connection are dropped. In addition, the TOE also controls that the SSHv2 session does not transmit more than 228 packets (less than 1GB) or that the duration of the session is more than one hour using the same session key. If the data transmission threshold is surpassed, or the session time limit is reached, new session keys are established between both ends.

The TOE creates session keys following the SSHv2 protocol specification and using the DRBG implemented in OpenSSL. The TOE destroys session keys when the session is terminated by clearing with zeroes and deallocating the Random Access Memory (RAM) memory used to store the session keys.

SSHv2 public and private keys are created by the TOE administrator via the CLI. Keys are also deleted from the filesystem by the TOE administrator using filesystem commands from the CLI (e.g. rm). The TOE removes from the filesystem the file entry corresponding to the key.

For destruction of plaintext keys in volatile storage, the TOE relies on the functionality provided by OpenSSL to zeroize and release the memory allocated for cryptographic operations. For keys in non-volatile storage, the TOE provides CLI commands to remove the relationship between keys and certificates and their usage in the TOE.

The following table shows the cryptographic keys involved in the SSHv2 protocol, their location and how they are created and destroyed.

Key	KeyLocation	Purpose	Destruction
SSH host RSA public key	/flash/system/ssh_host_rsa_key.pub	Key Establishment	Removal from filesystem through CLI commands
SSH host RSA private key	/flash/system/ssh_host_rsa_key	Key Establishment	
SSH host ECDSA private key	/flash/system/ssh_host_ecdsa_key	Key Establishment	
Session keys (shared secrets, encryption keys, data authentication keys)	RAM	Integrity and confidentiality during session	Zeroization and deallocation when session terminates

Table 6-7 Keys used by the SSHv2 protocol

6.2.4 Summary

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP22e:FCS_CKM.1: The TOE supports asymmetric key generation schemes using RSA, ECC and FFC safe-primes. The TOE generates RSA and ECDSA asymmetric cryptographic keys that are used to protect communications for TLSv1.2 and SSHv2. The TOE acts as a client for TLS generating RSA and ECC keys during key exchanges and a server for SSH generating RSA, ECC and FFC Safe-prime during key exchanges. RSA and ECDSA keys can also be generated during the creation of a Certificate Signing Request (CSR) or for use as an SSH host key.
- NDcPP22e:FCS_CKM.2: The TOE performs key establishment based on RSA and ECDSA asymmetric cryptographic keys that are used to protect communications for TLSv1.2 and SSHv2.
- NDcPP22e:FCS_CKM.4: The TOE destroys key material by overwriting it with zeroes and releasing the allocated memory only after it was properly destroyed.
- NDcPP22e:FCS_COP.1/DataEncryption: The TOE performs encryption and decryption using AES in CBC, CTR, and GCM mode with key sizes of either 128 or 256.
- NDcPP22e:FCS_COP.1/Hash: The TOE supports cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 with message digest sizes 160, 256, 384, and 512. Hashing is used within several services including TLS and SSH. SHA-256 is used to store passwords and in conjunction with published hases for verification of software image integrity.
- NDcPP22e:FCS_COP.1/KeyedHash: The TOE implements HMAC keyed hashing algorithm with SHA-1, SHA-256 and SHA-512 in accordance to these SFRs.
- NDcPP22e:FCS_COP.1/SigGen: The TOE supports the use of RSA with 2048 and 3072 bit key sizes, and ECDSA with a key size of 256 bits or greater for cryptographic signatures (specifically NIST curves P-256, P-384, or P-521).
- NDcPP22e:FCS_RBG_EXT.1: The TOE implements DRBG algorithms in accordance to this SFR. The TOE provides a software-based NDRNG based on HAVEGE (Hardware Volatile Entropy Gathering and Expansion) for seeding each DRBG with a minimum of 256 bits.

- NDcPP22e:FCS_SSHS_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above. The TOE supports authentication methods using either passwords or public-keys.
- NDcPP22e:FCS_TLSC_EXT.1/2: The TOE supports TLSv1.2 when exporting audit logs to an external server, and when communicating with authentication servers. The TOE is capable of providing a certificate in the TLS negotiation when the TLS server requests a certificate.

6.3 Identification and authentication

The TOE provides local and remote access to administrators; local access is provided through the serial console (connected to the available ports), whereas remote access is provided through the SSHv2 protocol using an SSH client. Both local and remote sessions can be terminated by the administrator at any time.

Before a local or remote session is established, a banner is displayed to the user that attempts to log into the TOE. An administrator of the TOE can modify the content of this banner to display warnings or advisory notices that reflect the security policy of the organization.

The TOE requires the administrator to identify and authenticate to the TOE prior to accessing any of the management functionality, regardless of the mechanism being used to interface with the TOE (e.g. serial console, SSH).

The TOE is shipped with a predefined user account: admin. The admin user account is the initial administrator assigned all privileges. This admin account can login at either the console or via SSH and is subject to be locked due to exceeding the lockout threshold through the SSH interface. However, the admin user account cannot be locked due to failing authentication attempts on the local console. The admin user account can login via the console even if the account is locked out at the SSH interface.

The TOE can perform identification and authentication locally using its local database. A successful logon occurs when the user presents to the TOE a valid administrative user name along with either the correct password or verification of a public-key. Once identification and authentication succeeds with the credentials provided by the user, the TOE grants access to the user by showing the command line interface (CLI) prompt.

6.3.1 Local Authentication

When configured for local authentication, the TOE maintains administrative-user security attributes of identifier (user ID), password information (authentication data), and user privileges (authorizations or user profile) and roles. The authentication data (password) is hashed prior to being stored using SHA-256. These attributes are stored locally on the flash file system, in directories protected from read and write access from the administration console.

If the user and password information entered by the user match the authentication data, authentication succeeds and the TOE grants access to the user.

The TOE provides the following user authentication failure settings

- Lockout window: The length of time a failed user login attempt is aged before it is no longer counted as a failed user login attempt. The valid range is 0 to 99,999. The number of failed login attempts is decremented by the number of failed attempts that age beyond the lockout window. The default lockout window is set to 0, which means that all consecutive failed login attempts are counted, regardless of how much time has elapsed between the failed logins.
- Lockout threshold: The number of failed user login attempts allowed within a given lockout window period of time (1-999). The default lockout threshold is set to 0, which means that there is no limit to the number of failed login attempts allowed, but this value is not allowed in the evaluated configuration.
- Lockout duration: The length of time a user account remains locked out until it is automatically unlocked. The valid range is 0 to 99,999. The default lockout duration is set to 0, which means that there is no automatic unlocking of a user account by the TOE.

The TOE ensures that if the number of failed user login attempts exceeds the lockout threshold during the lockout window period of time, the user account is locked out of the TOE for the lockout duration. The user is unlocked when either of the following conditions is met.

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

In either case, the user's authentication failure counter is reset when the user successfully authenticates.

The TOE provides global password settings used to implement and enforce local password complexity when a password is created or modified. The password settings available on the TOE are as follows:

- Minimum Password Length: The number of characters required when configuring a user password. The default value is 15 characters and can be changed within the range of 1 to 30 characters.
- Password Expiration: The number of days before user passwords will expire. The allowed range is 1-150 days. Password expiration is disabled by default.
- Username not allowed: Specifies whether or not the password is allowed to contain the username. The default is to allow the password to contain the username.
- Minimum Uppercase characters: Specifies the minimum number of uppercase characters required for a user password. The allowed range is 0-7. By default, there is no required minimum number of uppercase characters.
- Minimum Lowercase characters: Specifies the minimum number of lowercase characters required for a user password. The allowed range is 0-7. By default, there is no required minimum number of lowercase characters.
- Minimum Numeric characters: Specifies the minimum number of numeric characters (base-10 digits) required for a user password. The allowed range is 0-7. By default, there is no required minimum number of numeric characters.
- Minimum non-alpha characters: Specifies the minimum number of non-alphanumeric characters (symbols) required for a user password. The allowed range is 0-7. By default, there is no required minimum number of non-alpha characters.
- Password History: Specifies the maximum number of old passwords to retain. The range is 0-24 and the default is to retain 4 old passwords. The user is prevented from reusing any retained passwords. A value of 0 disables the password history function.
- Minimum Password Age: Specifies the minimum number of days during which the user is prevented from changing a password. The allowed range is 0-150. By default, there is no required minimum number of days.

When authentication is performed through a local session, the TOE does not display the password characters; instead, an asterisk is echoed for each character input.

6.3.2 X.509 Certificate Generation and Validation

The TOE supports X.509 certificate validation and certificate path validation according to [RFC5280]. They are used during the TLS handshake procedure to verify trust on the certificate received from the external IT entity, and verify the trust of the OCSP responder (if applicable).

When acting as a TLS client, the TOE parses and validates the TLS server certificate. If mutual authentication is required, the TOE sends the certificate used for that purpose that is located at the certificate directory (/flash/switch/cert.d).

Certificate validation and certificate path validation consist of the following steps.

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 ("Client Authentication", "Server Authentication", "OCSP Signing" purpose)

If all these steps are successful, then the certificate is considered valid. If any of these steps fails, then the certificate is considered invalid. Certificate pinning is not supported by the TOE.

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

The TOE contains a default CA keystore located in the flash filesystem (/flash/switch/ca.d). This keystore is used to perform certificate validation and contains all CA root certificates that are trusted by the TOE.

The TOE includes commands in the CLI to generate a Certificate Signing Request (CSR) and receive the corresponding CA certificate response file. After the CSR file is generated by the TOE, the administrator sends the request to a CA for being signed. Once the CA certificate response is received, the administrator uses the CLI to validate the certificate chain and import the signed certificate into the TOE. The TOE can create RSA-based as well as ECDSA-based CSR and import certificates built from these CSR into the TOE using the CLI.

The TOE provides the ***certificate delete*** command in the CLI to remove certificates and associated key from the filesystem.

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

- NDcPP22e:FIA_AFL.1: The TOE prevents the establishment of a remote session after a defined number of unsuccessful attempts using the password-based authentication method.
- NDcPP22e:FIA_PMG_EXT.1: The TOE allows the configuration of a password policy that includes a minimum length, and the combination of upper and lower case, numbers and special characters. The set of supported special characters is: '@', '#', '$', '%', '^', '&', '*', '(', ')', '~', '[', ']', ':', ';', '|', '_', '/', '.', '<', and '>'.
- NDcPP22e:FIA_UAU.7: The TSF does not echo any characters back to the local console while the user is entering their password.
- NDcPP22e:FIA_UAU_EXT.2: The TOE provides a password-based authentication mechanism.
- NDcPP22e:FIA_UIA_EXT.1: The TOE displays a warning banner before an administrative user attempts to login through a local (serial) or remote (SSHv2 client) console.
- NDcPP22e:FIA_X509_EXT.1/Rev: The TOE performs X.509 certificate validation using the Online Certificate Status Protocol (OCSP), as specified in [RFC6960].
- NDcPP22e:FIA_X509_EXT.2: The TOE supports X.509 certificate validation for the TLSv1.2 protocols.
- NDcPP22e:FIA_X509_EXT.3: The TOE allows the administrator to generate CSRs which contain the Common Name, Organization, Organizational Unit, and Country.

6.4 Security management

The TOE provides the following management functions for use by security administrators.

- Login at the local console or via SSH
- Configure the access banner for local and remote login sessions
- Configure session inactivity time for login sessions
- Install TOE firmware updates with the capability of verifying the integrity of those updates
- Configure authenticate failure parameters for login sessions (e.g. unsuccessful authentication attempts)
- Configure user login attempt lockout settings
- Configure audit behavior
- Manage cryptographic keys
- Configure cryptographic functionality
- Re-enable an Administrator account
- Set the date and time

- Manage the trust store and X.509v3 certificates (designate certificates as trust anchors, import X.509v3 certificates)

The security management functions can be performed through the Command Line Interface (CLI). The CLI can be accessed from the serial console or through a SSHv2 client.

Acting on behalf of a security administrator, the CLI requests security management operations from the same underlying service in the TOE. Therefore, although there are different methods of use for requesting the security management functions, each method utilizes the same underlying software to actually perform the functions.

Use of each of these management functions is restricted to the authorized administrator by requiring the administrator to successfully identify and authenticate to the TOE prior to allowing access to the functions. Administrators are granted access to management functions based on the access granted to their user account. The TOE provides the ability to grant read-only or read-write access to the command families available on the TOE. Examples of command families include file, system, config, module, interface, ip, vlan, dns, qos, policy, session, aaa. The aaa command family provides the ability to configure the type of authentication methods supported by the TOE and perform user account management.

Access to the trust store and X509.v3 certificates is restricted to administrators via the filesystem access control policies enforced by the TOE. Certificates are stored in the /flash/switch/cert.d directory, which acts as a keystore. Access to that directory is restricted to administrators.

The TOE restricts security administrators to determine and modify the behavior of security functions. No functions are available to any user before being identified and authenticated. Only the authorized administrator can configure TSF-related functions.

The Security management function satisfies the following security functional requirements:

- NDcPP22e:FMT_MOF.1/ManualUpdate: Only the authorized administrator can update the TOE.
- NDcPP22e:FMT_MTD.1/CoreData: No functions are available to any user before being identified and authenticated. Only the authorized administrator can configure TSF-related functions.
- NDcPP22e:FMT_MTD.1/CryptoKeys: Only the authorized administrator can configure cryptographic keys through commands on the CLI. The keys an authorized administrator can manage consist of importing trusted Root CA certs, generating SSH host keys, importing SSH public keys, and loading X.509 certificates. All of these keys can be also be deleted.
- NDcPP22e:FMT_SMF.1: The TOE allows the administrator to perform the administrative functions identified above.
- NDcPP22e:FMT_SMR.2: The TOE maintains administrative user roles which allow administrative accounts to have read-only or read-write permission to command families. Any administrative user that can login to the TOE, regardless of permission, is considered a security administrator.

6.5 Protection of the TSF

Passwords are stored in non-plaintext form using a hashing algorithm: SHA-256. The hashed value of the password is stored in a directory of the flash filesystem protected from read and write access.

The TOE includes the OpenSSL Cryptographic Library, which supports the TLS protocol and the underlying cryptographic algorithms used by the TOE. The module performs power-on self-tests (POST) to ensure the integrity of the module itself and the correct behavior of the cryptographic algorithms, and conditional tests to ensure that asymmetric key pairs are correctly generated. The POST proceeds until all self-tests are completed. In case of a failure in any of the self-tests, the TOE writes an error message to the console, and is forced to reload and reinitialize the operating system, thus performing the POST again. This mechanism ensures that no cryptographic algorithm is available until all self-tests are successful. Please refer to section 6.2 for a detailed description of self-tests and error messages that the module shows when the self-tests fail.

The TOE prevents access to all pre-shared keys, symmetric keys and private keys from the CLI by using the operating system's file access control: access to directories containing files with sensitive material is denied for all configured administrative users. The admin user, which is the default user in the TOE, is the only user that can have access to the files but its use is restricted to the installation and the initial configuration of the TOE.

The TOE does not provide a mechanism for automatic updates of the TOE. The administrator is responsible of following the instructions included in the TOE guidance to securely download, and install and/or update the TOE.

The TOE provides via the CLI the following commands for installing and updating the TOE in a secure way.

- Download the zip file containing both the new version of the TOE and its corresponding SHA-256 hash value (firmware image and hash value files) from the vendor's secure website
- Visualize the currently active version of the TOE, and the most recently installed version of the TOE (show microcode command)
- Verify the integrity of the downloaded image file that represents the TOE firmware against the SHA-256 hash value contained in a hash file (image integrity-check command.

When the TOE firmware is reloaded (reload from command), if the integrity of the TOE image is verified successfully, the command can proceed and the new version of the TOE is installed. If the integrity verification fails, then the TOE image is rejected and the command does not proceed with the update.

The TOE does not provide a means for installing a trusted update of the TOE with a delayed activation. However, the administrator must reboot the TOE in order to make the changes effective.

The TOE provides a reliable date and time for the following security functions.

- Generation of a timestamp for audit events
- Verification of the expiration of the certificate in X.509 certificate validation
- Calculation of period of inactivity of an interactive session to evaluate the termination of local and remote sessions

The TOE obtains the date and time from an internal system clock. This system clock can be updated by the security administrator through the CLI.

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

- NDcPP22e:FPT_APW_EXT.1: The TOE stores the hashed value of the password in a directory of the flash filesystem that is protected from read/access from any user, including administrators. The TOE does not offer any functions that will disclose to any user a plain text password.
- NDcPP22e:FPT_SKP_EXT.1: The TOE stores key material in directories of the flash filesystem which are protected from read/access from any user, including administrators. The TOE does not offer any functions that will disclose to any users a stored cryptographic key.
- NDcPP22e:FPT_STM_EXT.1: The TOE has an internal system clock which is used to generate reliable timestamps for security related purposes like auditing and certificate validation.
- NDcPP22e:FPT_TST_EXT.1: The TOE performs self-tests during start-up and conditional tests, which ensures the integrity of the Cryptographic Library and the correct operation of the cryptographic algorithms. The TOE also performs a verification of TOE executable using published hash during a startup.
- NDcPP22e:FPT_TUD_EXT.1: The TOE allows administrators to verify the executing version and the most recently installed version of the TOE software. It also allows manual updates of the TOE software, verifying its trust and integrity using a published hash before being installed.

6.6 TOE access

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

The TOE access function satisfies the following security functional requirements:

- NDcPP22e:FTA_SSL.3: The TOE terminates remote sessions after a configurable period of inactivity. After termination, administrative authentication is required to access any of the administrative functionality of the TOE.
- NDcPP22e:FTA_SSL.4: The TOE provides the function to logout (or terminate) both local and remote user sessions as directed by the user.
- NDcPP22e:FTA_SSL_EXT.1: The TOE terminates local sessions after a configurable period of inactivity. After termination, administrative authentication is required to access any of the administrative functionality of the TOE.
- NDcPP22e:FTA_TAB.1: The TOE allows the configuration of a banner shown to the user before an interactive session is established at the local console or remotely via SSH.

6.7 Trusted path/channels

The TOE provides a trusted path for its remote administrative users accessing the TOE using SSH. Note that local administrator access is also allowed for command line access.

Remote connections to third-party syslog servers are supported for exporting audit records to an external audit server. Administrators can configure communication with an external audit server to be protected using TLS.

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

- NDcPP22e:FTP_ITC.1: The TOE is a TLS client when when exporting audit records to a third party syslog server.
- NDcPP22e:FTP_TRP.1/Admin: All communications initiated by remote administrators to the TOE are protected using a secure channel via the SSHv2 protocol.