Arista Networks Switches Running EOS

Common Criteria Security Target

Version: 2.9

12/03/2019

Prepared By:

Arista Networks, Inc.
5453 Great America Parkway
Santa Clara, CA 95054

Table of Contents

1. Security Target (ST) Introduction

1.1 Security Target Reference

The Security Target reference uniquely identifies the Security Target.

ST Title:	Arista Networks Switches Running EOS
ST Version Number:	Version 2.9
ST Date:	3/Dec/2019
Keywords:	Network Device

1.2 Target of Evaluation Reference

The Target of Evaluation reference identifies the Target of Evaluation (TOE).

TOE Name:	Arista Networks Switches Running EOS
TOE Developer:	Arista Networks, Inc. 5453 Great America Parkway Santa Clara, CA 95054
TOE Software Version:	EOS 4.22.1FX-CC
TOE Hardware:	7500R, 7320X, 7300X, 7300X3, 7280R, 7260X, 7260X3, 7250QX, 7170, 7160, 7060X, 7060X4, 7050X3, 7050X, 7020R, 7010T and 720XP series switches.
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.
PP Identification:	collaborative Protection Profile for Network Devices, Version 2.1, September 24, 2018.

1.3 Target of Evaluation Overview

1.3.1 TOE Type

The TOE is classified as a Network Device, that is, a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network.

1.3.2 TOE Usage

The Arista Networks Data Center and Cloud Computing Switches are networking switches (Network Devices for CC purposes) that provide OSI model Layer 2, 3, and 4 Ethernet interconnectivity and network management services (Data Link, Network, and Transport Layers, respectively). Each model is manufactured with high performance electronics making it ideally suitable for demanding data center environments.

1.3.3 TOE Major Security Features Summary

- Security Audit

- Generates audit records, storing them locally and transmitting them to a remote audit server.
- Supports secure communication to remote syslog-compatible audit servers protected by the SSHv2 Trusted Channel.

- Cryptographic Support ○ Utilization of NIST-specified and CAVP validated cryptographic algorithms for asymmetric key generation, AES encryption/decryption, digital signature generation/verification, hashing, and keyed-hashing (Message Authentication Code).
  - Cryptographic-key Destruction using PP specified methods. ○ Deterministic Random Bit Generation (DRBG).
  - Assurance of seeding the DRBG with sufficient entropy (minimum of 256-bits of entropy).

- Identification and Authentication ○ Administrative password management. ○ Protected authentication data at the local and remote consoles. ○ Identification and authentication of the Security Administrative user. ○ X509 certificate based authentication and validation. ○ X509 certificate request generation.

- Security Management ○ Trusted Update mechanism. ○ Restriction of TSF management to the Security Administrator.
  - Local and remote administration of the TOE by the Security Administrator.

- Protection of the TSF ○ Protection of stored passwords. ○ Prevention of disclosing passwords via normal management interfaces. ○ Prevention of disclosing private keys via normal management interfaces. ○ Automated self-testing upon boot-up. ○ Querying of the TOE firmware/software. ○ Reliable timestamps.

- TOE Access
  - Session termination (TSF initiated and User initiated).
  - Display of a warning and consent banner on the local and remote management interfaces prior to authentication.

- Trusted Path/Channels ○ Cryptographically secure path between the TOE and the Security Administrative user for remote management. ○ Cryptographically secure channels between the TOE and authorized IT entities in the Operational Environment to support the TSF.

1.3.4 Operational Environment

The TOE's Operational Environment must provide the following services to support the secure operation of the TOE:

- Local Console Administrative Access ○ RS-232 Serial Console. ○ VT-100 terminal emulation program.

- Remote Management
  - SSH client for remote interactive session utilizing SSH.

- eAPI JSON-RPC Client capable of establishing a mutually authenticated TLS session.

- Audit Server ○ Syslog server capable of accepting an SSHv2 tunnel utilizing SSH Protocol Version 2 (SSHv2).
- Certificate Revocation List (CRL) Server ○ Server from where CRLs can be downloaded on TOE to check validity of X509v3 certificates.

1.4 Target of Evaluation Description

The Arista 7500R, 7320X, 7300X and 7300X3 series switches have modular architecture. The modular switch consists of a base chassis, which is a housing with spaces for insertion of different modules of the switch. Chassis range in sizes between 7 and 21 RU. Following modules are required to be inserted in the chassis to make the switch operational: Power supply, supervisor module, line card and fabric module. Supervisor module performs the central control and management functions. Line card provides copper/optical network ports and high-speed traffic forwarding plane among them. Multiple line cards can be added depending on port number requirements. Fabric module interconnects lines cards and supervisor module.

The Arista 7280R, 7260X, 7260X3, 7250QX, 7170, 7160, 7060X, 7060X4, 7050X3, 7050X, 7020R, 7010T and 720XP Series Switches are fixed form factor switches. The fixed form factor switches range in size between 1 and 2 RU. In each series, different models vary in total throughput, port count, port speeds, route table scales etc.

Each switch model runs Arista's Linux-based network operating system called Extensible Operating System (EOS). The same EOS binary image runs on all TOE hardware models. For modular switches, the EOS runs on supervisor module. For the fixed form factor switches, it runs on host CPU in them. All EOS code is compiled to the same i686 assembly, making it such that no processor runs anything different from any other processor. All processors implement the i686 assembly language. All SFRs in this Security Target are implemented by EOS. Hence, they behave identically on every switch model.

The table below provides the list of appliances across different series:

Table 1: Hardware Appliances

Series	Models	Supervisor/Host CPU
7500R	Chassis Options: ● DCS-7504, DCS-7504N with 4 line card slots ● DCS-7508, DCS-7508N with 8 line card slots	Intel Broadwell-DE

	● DCS-7512, DCS-7512N with 12 line card slots ● DCS-7516, DCS-7516N with 16 line card slots <u>Supervisor Module Options:</u> ● DCS-7500-SUP2 ● DCS-7516-SUP2 <u>Line Card Options:</u> ● DCS-7500R-36CQ (36x QSFP100 100G ports) ● DCS-7500R-36Q (36x QSFP+ 40G ports) ● DCS-7500R-48S2CQ (48x SFP+ 10G & 2x QSFP100 100G ports) ● DCS-7500R2A-36CQ, 7500R2-36CQ (36x QSFP100 100G ports) ● DCS-7500R2AK-36CQ (36x QSFP100 100G ports) ● DCS-7500R2AK-48YCQ (48x SFP+ 10G & 2x QSFP100 100G ports)	
7320X, 7300X, 7300X3	<u>Chassis Options:</u> ● DCS-7304 with 4 line card slots ● DCS-7308 with 8 line card slots ● DCS-7316 with 16 line card slots <u>Supervisor Module:</u> ● DCS-7300-SUP <u>Line Card Options:</u> ● DCS-7320X-32C-LC (32x QSFP100 100G ports) ● DCS-7300X-32Q-LC (32x QSFP+ 40G ports) ● DCS-7300X-64S-LC (48x SFP+ 10G & 4x QSFP+ 40G ports) ● DCS-7300X-64T-LC (48x 10GBASE-T 10G & 4x QSFP+ 40G ports)	Intel Sandy Bridge EN

7280R (Subseries: 7280CR)	● 7280CR2A-30, 7280CR2K-30 (30x QSFP100 100G ports)	Intel Sandy Bridge EN
	● 7280CR-48 (48x QSFP100 100G & 8x QSFP+ 40G ports) ● 7280CR2-60, 7280CR2A-60, 7280CR2K-60 (60x QSFP100 100G ports)	
7280R (Subseries: 7280QR)	● 7280QR-C36, 7280QRA-C36S (24x QSFP+ 40G & 12x QSFP100 100G ports) ● 7280QR-C72 (56x QSFP+ 40G & 16x QSFP100 100G ports)	AMD G Series: Steppe Eagle
7280R (Subseries: 7280SR)	● 7280SRA-48C6, 7280SR-48C6 (48x SFP+ 10G & 6x QSFP100 100G ports) ● 7280SR2A-48YC6, 7280SR2-48YC6 (48x SFP25 25G & 6x QSFP100 100G ports) ● 7280SR2K-48C6 (24x SFP+ 10G & 24x SFP25 25G & 6x QSFP100 100G ports)	AMD G Series: Steppe Eagle
7280R (Subseries: 7280TR)	● 7280TRA-48C6, 7280TR-48C6 (48x 10GBASE-T 10G & 6x QSFP100 100G ports)	AMD G Series: Steppe Eagle
7260X	● 7260CX-64, 7260QX-64 (64x QSFP+ 40G ports)	Intel Sandy Bridge EN or AMD G Series: eKabini
7260X3	● 7260CX3-64 (64x QSFP100 100G ports)	Intel Broadwell-DE
7250QX	● 7250QX-64 (64x QSFP+ 40G ports)	Intel Sandy Bridge EN
7170	● 7170-32C (32x QSFP100 100G ports) ● 7170-64C (64 x QSFP100 100G ports)	Intel Broadwell-DE
7160	● 7160-32CQ (32x QSFP100 100G ports) ● 7160-48TC6 (48x SFP25 25G & 6x QSFP100 100G ports) ● 7160-48YC6 (48x 10GBASE-T 10G & 6: QSFP100 100G ports)	AMD G Series: eKabini or AMD G Series: Steppe Eagle
7060X	● 7060CX2-32S, 7060CX-32S (32x QSFP100 100G & 2x SFP+ 10G ports) ● 7060SX2-48YC6 (48 x SFP25 25G & 6x QSFP100 100G ports)	AMD G Series: Steppe Eagle

7060X4	● 7060PX4-32 (32x OSFP 400G ports & 2x SFP+ 10G ports) ● 7060DX4-32 (32x QSFP-DD 400G ports & 2x SFP+ 10G ports)	Intel Broadwell-DE
7050X3	● 7050CX3-32S (32x QSFP100 100G ports) ● 7050SX3-48YC12 (48x SFP25 25G & 12x QSFP100 100G ports)	AMD G Series: Steppe Eagle
7050X (Subseries: TX, SX, QX)	● 7050TX-48 (32x 10GBASE-T 10G & 4x QSFP+ 40G ports)	AMD G Series: eKabini
	● 7050TX-64 (48x 10GBASE-T 10G & 4x QSFP+ 40G ports) ● 7050TX-72 (48x 10GBASE-T 10G & 2x MXP ports) ● 7050TX-96 (48x 10GBASE-T 10G & 4x MXP ports) ● 7050TX-128 (96x 10GBASE-T 10G & 8x QSFP+ 40G ports) ● 7050TX2-128 (96x 10GBASE-T 10G & 8x QSFP+ 40G ports) ● 7050SX-64 (48x SFP+ 10G & 4x QSFP+ 40G ports) ● 7050SX-72 (48x SFP+ 10G & 2x MXP ports) ● 7050SX-96 (48x SFP+ 10G & 4x MXP ports) ● 7050SX2-128 (96x 10GBASE-T 10G & 8x QSFP+ 40G ports) ● 7050QX-32S, 7050QX2-32S (32x QSFP+ 40G & 4x SFP+ 10G ports)	
7050X (Subseries: TX, SX)	● 7050TX-72Q (48x 10GBASE-T 10G & 6x QSFP+ 40G ports) ● 7050SX-72Q, 7050SX2-72Q (48x SFP+ 10G & 6x QSFP+ 40G ports)	AMD G Series: Steppe Eagle
7050X (Subseries: SX)	● 7050SX-128 (96x SFP+ 10G & 8x QSFP+ 40G ports)	Intel Sandy Bridge EN

7020R	• 7020SR-24C2, 7020SRG-24C2 (24x 10GBASE-T 10G & 2x QSFP100 100G ports) • 7020TR-48, 7020TRA-48 (48x 1000BASE-T 1G & 6x SFP+ 10G ports)	AMD G Series: Steppe Eagle
7010T	• 7010T-48 (48x 1000BASE-T 1G & 4x SFP+ 10G ports)	AMD G Series: eKabini
720XP	• 720XP-48ZC2-F (40x 2.5G & 8x 5G ports) • 720XP-24ZY4-F (16x 2.5G & 8x 5G ports) • 720XP-48Y6-F (48x 1G ports) • 720XP-24Y6-F (24x 1G ports)	AMD G Series: Steppe Eagle

The TOE supports local administration via the local console port. Remote administration is performed over the Secure Shell v2 (SSHv2) protocol. Alternatively, management of the TSF can be automated and performed remotely over TLS connection via the eAPI automated remote management interface ("eAPI") using the eAPI JSON-RPC Client.

The TOE also supports storage and forwarding of audit records, protected using SSHv2, to any syslog-compatible network entity.

1.4.1 Target of Evaluation Architecture

EOS includes subsystems designed to implement operational, security, management and networking functions. In the modular switch, the necessary hardware (CPU, Flash, RAM, serial and network interfaces) for EOS to run resides in the supervisor module. In the fixed form factor switch, this hardware is included in the switch appliance. EOS contains management interface subsystem comprising of applications that implement Serial Console, eAPI and SSH. This subsystem utilizes APIs provided by the Crypto Module to implement cryptographic algorithms. The keys and certificates database supports operation of cryptographic algorithms. The AAA subsystem maintains administrative user credentials, which the management subsystem relies on to identify and authenticate the users. The Audit Agent creates audit logs on relevant events, sends them to remote audit server utilizing the services of SSH, and stores them on local storage. The Config Database stores switch configuration. The Switching and Routing Engine performs core function of the switch, which is to implement traffic forwarding logic. The rules generated by this engine are programmed into line cards, which perform actual traffic forwarding function. In the modular switch, line cards communicates to supervisor module through fabric modules. In the fixed form factor switch, line cards communicate with the Switching and Routing Engine via communication bus inside the appliance.

The TOE architecture and subsystem interactions are shown in Figure 1.

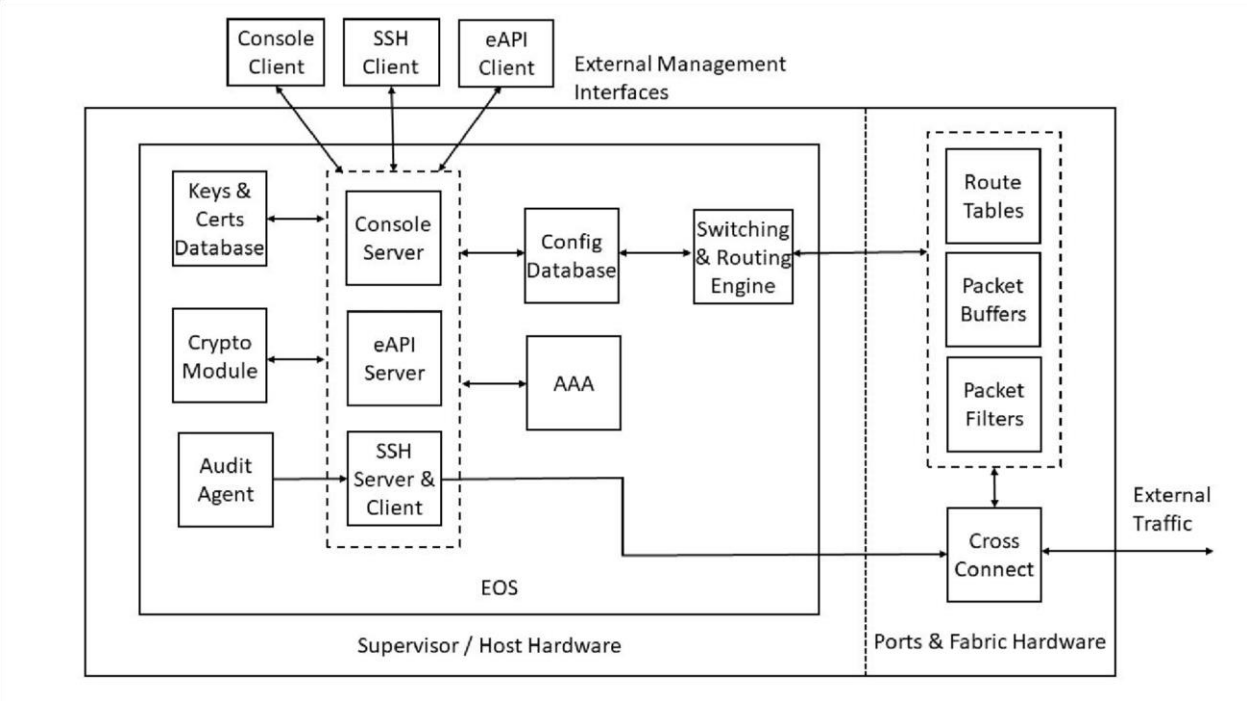

Figure 1: TOE Architecture

1.4.2 Target of Evaluation Physical Boundary

Physical boundary of the TOE is a switch appliance of one of the models described in Table 1, including all its hardware, firmware, software, local and remote management interfaces and Arista Extensible Operating System (EOS) version EOS 4.22.1FX-CC.

For the fixed form factor switches, the switch appliance contains host CPU, DRAM and flash to run EOS. There are fixed number of copper or optical network ports on the appliance. The physical boundary of the TOE is the switch appliance as shown in Figure 2.

Physical Boundary

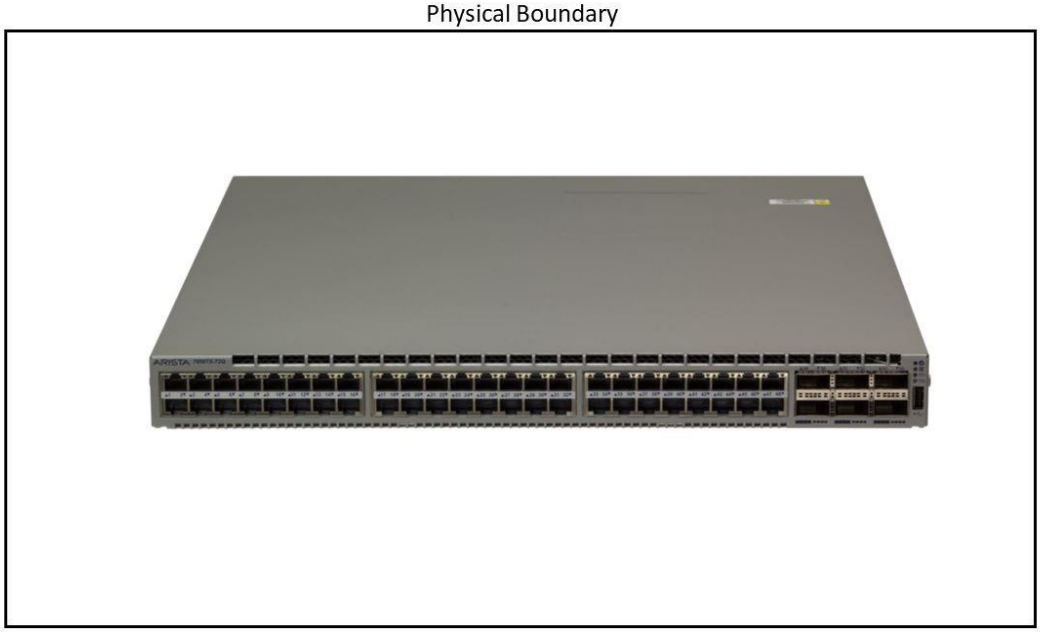

Figure 2: Physical Boundary of Fixed Form Factor Switch (7050TX-72Q)

For the modular switches, the physical boundary is a switch appliance consisting of a chassis and supervisor module(s), line card(s), fabric card(s) and power module(s) inserted into the chassis. The physical boundary is shown in Figure 3A and Figure 3B. The supervisor module contains supervisor CPU, DRAM and flash to run EOS. At least one supervisor module is required for the switch to function. Additional supervisor module can be added to provide redundancy. Line cards are expansion cards that provide copper or optical network ports. Appropriate number of line cards are added to meet the port number requirements. Fabric modules provide interconnectivity among line cards and supervisor modules.

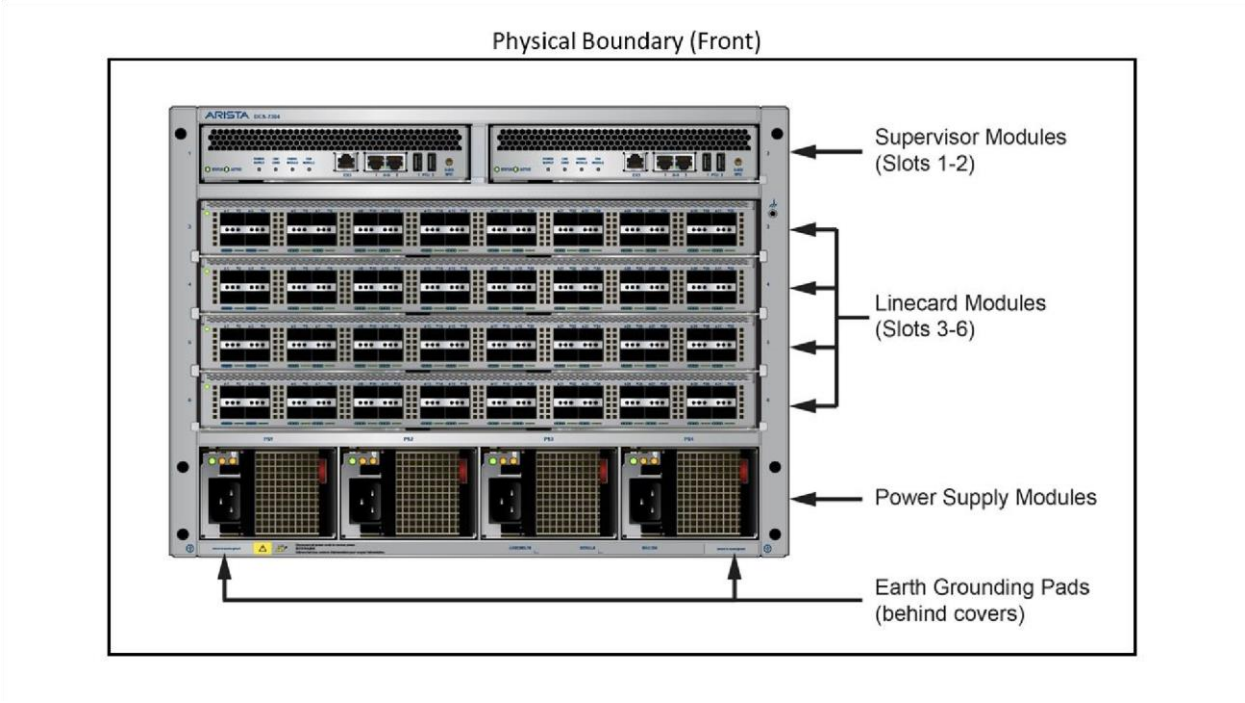

Figure 3A: Physical Boundary of Modular Switch (DCS-7304, Front View)

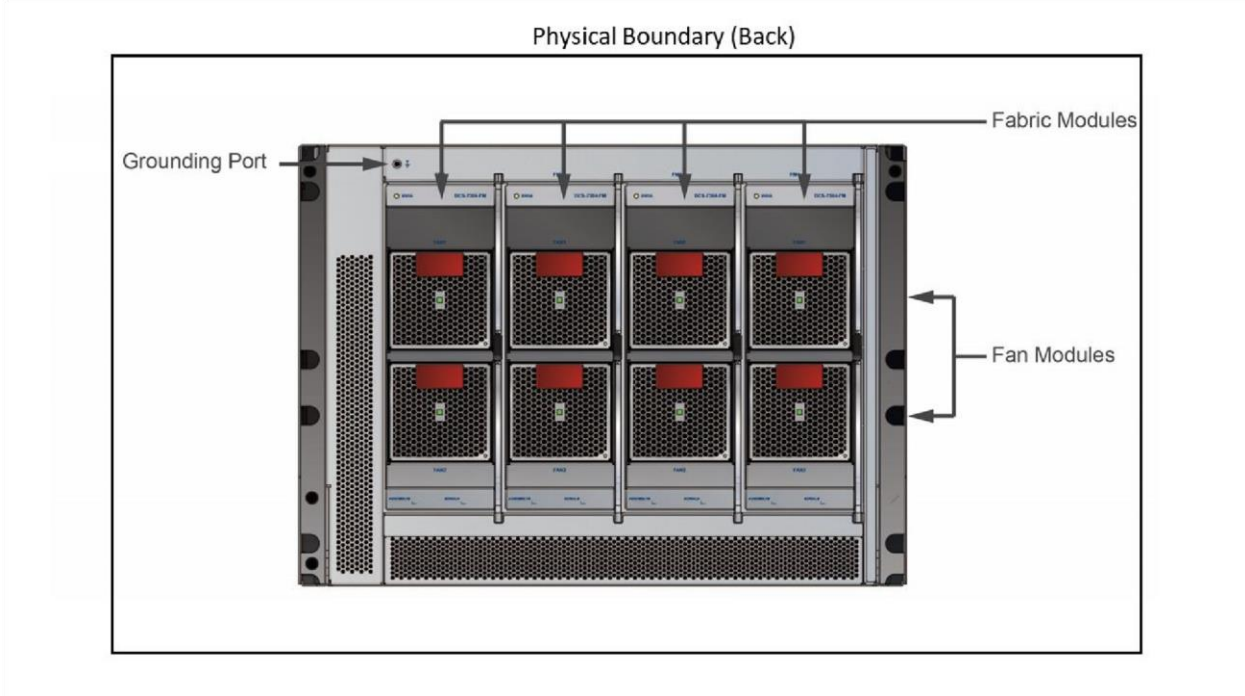

Figure 3B: Physical Boundary of Modular Switch (DCS-7304, Back View)

1.4.3 Target of Evaluation Logical Boundary

The logical boundary of the TOE includes the security functions implemented exclusively by the TOE. These security functions are summarized in Section 1.3.3 above and are further described throughout the

security target. A more detailed description of the implementation of these security functions are provided in Section 7 “TOE Summary Specification”.

1.4.3.1 Security Audit

- The TOE will audit all events and information defined in Table 8.
- The TOE will also include the identity of the user that caused the event (if applicable), date and time of the event, type of event, and the outcome of the event.
- The TOE protects storage of audit information from unauthorized deletion.
- The TOE prevents unauthorized modifications to the stored audit records.
- The TOE can transmit audit data to an external IT entity using the SSHv2 protocol.

1.4.3.2 Cryptographic Support

The TOE implements CAVP validated cryptographic algorithms for asymmetric key generation, encryption/decryption, digital signature, integrity protection/verification and random bit generation. These algorithms are used to provide security for the SSH and TLS connections of the Trusted Path and Trusted Channel.

1.4.3.3 Identification and Authentication

- The TSF supports passwords consisting of alphanumeric and special characters. The TSF also allows administrators to set a minimum password length and support passwords of 15 characters or greater.
- The TSF requires all administrative-users to authenticate before allowing the user to perform any actions other than: ○ Viewing the warning banner.

1.4.3.4 Security Management

The TOE allows human users with the Security Administrator role to administer the TOE over a remote console (SSH Trusted Path) and local CLI (Local Console). The eAPI JSON-RPC trusted IT entity client allows machine users with the Security Administrator role to administer the TOE over a remote TLS Trusted Channel. These interfaces do not allow the Security Administrator to execute arbitrary commands or executables on the TOE.

1.4.3.5 Protection of the TSF

- The TSF prevents the reading of secret keys, private keys and passwords.
- The TOE runs a suite of self-tests, during the initial start-up (upon power on), and when programs which utilize the cryptographic libraries are initialized, to demonstrate the correction operation of the TSF.
- The TOE provides a means to verify firmware/software updates to the TOE using a published hash prior to installing those updates.
- The TOE provides reliable time stamps for itself.

1.4.3.6 TOE Access

- The TOE, for local interactive sessions, terminates the session after Security Administrator-specified period of session inactivity.

- The TOE terminates a remote interactive session after Security

  Administrator-configurable period of session inactivity.
- The TOE allows Administrator-initiated termination of the Administrator's own interactive session.
- Before establishing an administrative user session, the TOE is capable of displaying Security Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

1.4.3.7 Trusted Path/Channels

- The TOE uses SSH or TLS to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and modification.
- The TOE permits the TSF, or the authorized IT entities to initiate communication via the trusted channel.
- The TOE permits remote administrators to initiate communication via the trusted path.
- The TOE requires the use of the trusted path for initial administrator authentication and all remote administration actions.

1.5 Excluded Functionality

The following features should not be used in the CC evaluated configuration. They are disabled by default (e.g., telnet) or require explicit additional configuration to make them work (e.g., integration with remote authentication server). These have not been evaluated.

- Telnet management interface.
- HTTP and HTTPS web GUI management interface.
- Integration with external authentication server over RADIUS and TACACS+.
- Management interfaces for XMPP, Openconfig, CloudVision eXchange (CVX) and CloudVision Portal (CVP).
- SNMP for management and notification.
- SMTP to post email notifications.
- Real-time streaming of switch state to remote server using `TerminAttr' service.
- Remote configuration backup with CLI command.
- FTP server.
- Integration with orchestration services such as Puppet, Ansible, Chef, Prometheu etc. by installing their agents on the switch.
- 1+1 redundant supervisor modules in modular switches.

The following features have also not been evaluated as their RFC-compliant implementations are unable to satisfy cryptographic requirements outlined in the PP.

- Routing protocols that integrate authentication or encryption such as Routing Information

Protocol (RIPv1, RIPv2), Open Shortest Path First (OSPFv2), Border Gateway Protocol (BGP), Intermediate System to Intermediate System (IS-IS), and Virtual Router Redundancy Protocol (VRRP).

In the evaluated configuration, the switch supports eAPI JSON-RPC interface over TLS for remote automation scripts to perform management functions on the switch. This interface supports only JSON request/response format. This is a machine-to-machine interface and not to be used as human interactive interface.

2. Conformance Claims

2.1 Common Criteria Conformance Claims

This Security Target is conformant to the Common Criteria Version 3.1 R5, CC Part 2 extended [C2], and CC Part 3 conformant [C3].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the collaborative Protection Profile for Network Devices, Version 2.1, dated September 24, 2018 [S1]. This Protection Profile will be referred to as cPP or PP throughout this Security Target.

2.3 Conformance to Technical Decisions

The TOE complies with the following Technical Decisions.

- 0402 – NIT Technical Decision for RSA-based FCS_CKM.2 Selection
- 0401 – NIT Technical Decision for Reliance on external servers to meet SFRs
- 0400 – NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment
- 0412 - NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy
- 0399 – NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)
- 0398 – NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR
- 0397 – NIT Technical Decision for Fixing AES-CTR Mode Tests
- 0395 – NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2 Note: Technical decision 0396 is not applicable to TOE.

2.4 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements or security assurance requirements packages, neither as package-conformant or package-augmented.

2.5 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
  - All threats defined in the cPP are carried forward to this ST; ○ No additional threats have been defined in this ST.
- Organizational Security Policies ○ All OSP defined in the cPP are carried forward to this ST;

○ No additional OSPs have been defined in this ST.

- Assumptions

○ All assumptions defined in the cPP are carried forward to this ST;
○ No additional assumptions for the operational environment have been defined in this ST.

- Objectives
  - All objectives defined in the cPP are carried forward to this ST.
- All mandatory and selection-based SFRs and SARs defined in the cPP are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the cPP have been properly instantiated in this Security Target; therefore, this ST shows exact compliance to the cPP.

3. Security Problem Definition

3.1 Threats

The following table defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP unchanged.

Table 2: Threats	
Threat	Description

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

T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.

T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access,
	change or modify device data, critical network traffic or security functionality of the device.

3.2 Organizational Security Policies

The following table defines the organizational security policies which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP unchanged.

Table 3: Organizational Security Policies	
OSP	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP unchanged.

Table 4: Assumptions	
Assumption	Description

A.PHYSICAL_PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the device should not provide computing platform for general purpose Applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTIO N	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is
	assumed that this protection will be covered by cPPs for particular types of network devices (e.g, firewall).

A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device. The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

4. Security Objectives

4.1 Security Objectives for the Operational Environment

Table 5: Security Objectives for the Operational Environment	
Objective	Description

OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5. Extended Components Definition

5.1 Extended Security Functional Components

The extended components listed in Table 6 along with their definitions have been sourced from cPPND v2.1.

Table 6: Functional Components

SFR	Description	Definition and Section from NDcPP v2.1
FAU_STG_EXT.1	Protected Audit Event Storage	Protected audit event storage requires

		the TSF to use a trusted channel implementing a secure protocol. (C.1.2)

FCS_RBG_EXT.1	Random Bit Generation	Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. (C.2.2.1)
FCS_SSHC_EXT.1 (selection based)	SSH Client Protocol	SSH Client requires that the client side of SSH be implemented as specified. (C.2.2.6)
FCS_SSHS_EXT.1 (selection based)	SSH Server Protocol	SSH Server requires that the server side of SSH be implemented as specified. (C.2.2.7)
FCS_TLSS_EXT.2 (selection based)	TLS Server Protocol with Mutual Authentication	TLS Server requires the mutual authentication be included in the TLS implementation (C.2.2.9)
FIA_PMG_EXT.1	Password Management	Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints. (C.3.3.1)
FIA_UIA_EXT.1	User Identification and Authentication	User Identification and Authentication requires Administrators (including remote Administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions (C.3.2)
FIA_UAU_EXT.2	Password-based Authentication Mechanism	The password-based authentication mechanism provides administrative users a locally based authentication mechanism (C.3.3)
FIA_X509_EXT.1/Rev (selection based)	X.509 Certificate Validation	X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component. (C.3.4)

FIA_X509_EXT.2 (selection based)	X.509 Certificate Authentication	X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates. (C.3.4)
FIA_X509_EXT.3 (selection based)	X.509 Certificate Requests	X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses. (C.3.4)
FPT_SKP_EXT.1	Protection of TSF Data (for reading all symmetric keys)	Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family. (C.4.1)
FPT_APW_EXT.1	Protection of Administrator Passwords	Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject. (C.4.2.1)
FPT_TST_EXT.1	TSF Testing	TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF. (C.4.3.1)
FPT_TUD_EXT.1	Trusted Update	Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation. (C.4.4)
FPT_STM_EXT.1	Reliable Time Stamps	Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps. (C.4.5)
FTA_SSL_EXT.1	TSF-initiated Session Locking	TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family. (C.5.1)

5.2 Extended Security Functional Requirements Rationale

All extended security functional components are sourced directly from cPP. Exact conformance required by the cPP also mandates inclusion of all applicable extended components defined in the cPP.

6. Security Requirements

6.1 Security Functional Requirements

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 number in parenthesis following the requirement number, e.g., FCS_COP.1.1(1).
  - **Assignment**: Allows the specification of an identified parameter. Assignments made by the ST author are identified with **bold text.**
  - **Selection**: Allows the specification of one or more elements from a list. Selections are identified with underlined text **.**
  - **Refinement**: Allows the addition of details. Additions to the CC text are specified in ***italicized bold and underlined text***. Refinements that add text use ***bold and italicized text*** to identify the added text**.** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

Note that operations already performed in the PP are not identified in this Security Target.

- **Explicitly stated Security Functional Requirements** (i.e., those not found in Part 2 of the CC) are identified "_EXT" in the component name.)

The TOE security functional requirements are listed in Table 6. All SFRs are based on requirements defined in Part 2 of the Common Criteria, or defined in cPP.

Table 7: Security Functional Requirements	
SFR	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation (Refined)
FCS_CKM.2	Cryptographic Key Establishment (Refined)

FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHC_EXT.1 (selection based)	SSH Client Protocol
FCS_SSHS_EXT.1 (selection based)	SSH Server Protocol
FCS_TLSS_EXT.2 (selection based)	TLS Server Protocol with Mutual Authentication
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev (selection based)	X.509 Certificate Validation
FIA_X509_EXT.2 (selection based)	X.509 Certificate Authentication
FIA_X509_EXT.3 (selection based)	X.509 Certificate Requests
FMT_MOF.1/ManualUpdate	Manual Update
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading all symmetric keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking

FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_TRP.1/Admin	Trusted Path

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
   - Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).
   - Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
   - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
   - Resetting passwords (name of related user account shall be logged).
   - no other actions;
d) Specifically defined auditable events listed in Table 2.

FAU_GEN.1.2

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 2.

Table 8: Auditable Events (Table 2 of cPP)		
SFR	Description	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.

FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_SSHC_EXT.1 (selection based)	Failure to establish an SSH session	Reason for failure.
		Non-TOE endpoint of connection (IP Address)
FCS_SSHS_EXT.1 (selection based)	Failure to establish an SSH session	Reason for failure.
FCS_TLSS_EXT.2 (selection based)	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FCS_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev (selection based)	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store.	Reason for failure of certificate validation. Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store.
FIA_X509_EXT.2 (selection based)	None.	None.
FIA_X509_EXT.3 (selection based)	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.

FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
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.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

6.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1

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

6.1.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

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 FTP_ITC.1.

FAU_STG_EXT.1.2

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

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

FAU_STG_EXT.1.3

The TSF shall overwrite previous audit records according to the following rule: **periodic audit log rotation (delete the oldest log file)** when the local storage space for audit data is full.____

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

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;
- FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3 ~~and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following:~~

~~[assignment: *list of standards*]~~.

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;

~~that meets the following: [assignment: *list of standards*]~~.

6.1.2.3 FCS_CKM.4: Cryptographic Key Destruction

FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a single overwrite consisting of zeroes;

- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that:
    - logically addresses the storage location of the key and performs a single-pass overwrite consisting of zeros

that meets the following: No Standard.

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

FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in CBC 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 .

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

FCS_COP.1.1/SigGen

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) **2048 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.

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

FCS_COP.1.1/Hash

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-1, SHA-256, SHA-512 and message digest sizes 160 , 256, 512 bits that meet the following: ISO/IEC 10118-3:2004.

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

FCS_COP.1.1/KeyedHash

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

6.1.2.8 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from **2** software-based noise source with a minimum of 256 bits of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

6.1.2.9 FCS_SSHC_EXT.1 SSH Client Protocol (Selection Based)

FCS_SSHC_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFC(s) 4251 , 4252, 4253, 4254, 6668, 8332.

FCS_SSHC_EXT.1.2

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

FCS_SSHC_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than **262,144** bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc .

FCS_SSHC_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses rsa-sha2-256 as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses hmac-sha2-256 as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7

The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8

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

FCS_SSHC_EXT.1.9

The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or no other methods as described in RFC 4251 section 4.1.

6.1.2.10 FCS_SSHS_EXT.1 SSH Server Protocol (Selection Based)

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251 , 4252, 4253, 4254, 6668, 8332.

FCS_SSHS_EXT.1.2

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

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than **262,144** bytes in an SSH transport connection are dropped.

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 .

FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses rsa-sha2-256 as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses hmac-sha2-256 as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange methods used for the SSH protocol.

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 no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

6.1.2.11 FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication (Selection Based)

FCS_TLSS_EXT.2.1

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

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

FCS_TLSS_EXT.2.2

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

FCS_TLSS_EXT.2.3

The TSF shall perform RSA key establishment with key size 2048 bits; generate Diffie-Hellman parameters of size 2048 bit.

FCS_TLSS_EXT.2.4

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

FCS_TLSS_EXT.2.5

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- Not implement any administrator override mechanism

].

FCS_TLSS_EXT.2.6

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

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within **1 to 255** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed.

6.1.3.2 FIA_PMG_EXT.1 Password Management

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 [S] hall be configurable to between **1** and **32** characters.

6.1.3.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1; ● no other actions.

FIA_UIA_EXT.1.2

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

6.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, none to perform administrative user authentication.

6.1.3.5 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

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

6.1.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation (Selection Based)

FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3.

- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. ○ Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP Certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2/Rev

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

6.1.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

6.1.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

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

FIA_X509_EXT.3.2

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

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

FMT_MOF.1.1/ManualUpdate

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

6.1.4.2 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData

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

6.1.4.3 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys

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

6.1.4.4 FMT_SMF.1 Specification of Management Functions

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 manage the cryptographic keys; ○ Ability to configure the cryptographic functionality; ○ 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 X.509v3 certificates to the TOE's trust store;.

6.1.4.5 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1

The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions:

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

are satisfied.

6.1.5 Protection of the TSF (FPT)

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

FPT_SKP_EXT.1.1

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

6.1.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext passwords.

6.1.5.3 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests during initial start-up (on power on), at the conditions **as specified by FIPS PUB 140-2 Section 4.9.2** to demonstrate the correct operation of the TSF: **Power-up self-tests: Integrity check of the cryptographic module and Known Answer Tests (KAT) of cryptographic primitives, and conditional self-tests: Key generation pairwise consistency tests and continuous random number generator testing**.

6.1.5.4 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and no other TOE firmware/software version .

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a published hash prior to installing those updates.

6.1.5.5 FPT_STM_EXT.1: Reliable Time Stamps

FPT_STM_EXT.1.1

The TSF shall be able to provide reliable time stamps.

FPT_STM_EXT.1.2

The TSF shall allow the Security Administrator to set the time.

6.1.6 TOE Access (FTA)

6.1.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions,

- terminate the session

after a Security Administrator-specified time period of inactivity.

6.1.6.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

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

6.1.6.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

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

6.1.6.4 FTA_TAB.1: Default TOE Access Banners

FTA_TAB.1.1

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

6.1.7 Trusted Path/Channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

The TSF shall be capable of using SSH, TLS to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, eAPI JSON-RPC Client that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for **transmitting audit records to audit server**.

6.1.7.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin

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

FTP_TRP.1.2/Admin

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

FTP_TRP.1.3/Admin

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

6.2 Security Assurance Requirements

6.2.1 Security Assurance Requirements for the TOE

This section defines the assurance requirements for the TOE. The assurance activities to be performed by the evaluator are defined in cPP and are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Table 9: Assurance Requirements

Assurance Class	Assurance Component
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing –sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

The following tables state the developer action elements, content and presentation elements and evaluator action elements for each of the assurance components.

Table 10: ADV_FSP.1 Basic Functional Specification	
Developer Action Elements	
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.
Content and Presentation Elements	
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.
Evaluator Action Elements	
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.

Table 11: AGD_OPE.1 Operational User Guidance
Developer Action Elements

AGD_OPE.1.1D	The developer shall provide operational user guidance.
Content and Presentation Elements	
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.
Evaluator Action Elements	
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Table 12: AGD_PRE.1 Preparative Procedures	
Developer Action Elements	
AGD_PRE.1.1D	The developer shall provide the TOE, including its preparative procedures.
Content and Presentation Elements	

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.
Evaluator Action Elements	
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.

Table 13: ALC_CMC.1 Labeling of the TOE	
Developer Action Elements	
ALC_CMC.1.1D	The developer shall provide the TOE and a reference for the TOE.
Content and Presentation Elements	
ALC_CMC.1.1C	The TOE shall be labeled with its unique reference.
Evaluator Action Elements	
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Table 14: ALC_CMS.1 TOE CM Coverage	
Developer Action Elements	
ALC_CMS.1.1D	The developer shall provide a configuration list for the TOE.

Content and Presentation Elements	
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.
Evaluator Action Elements	
ALC_CMS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Table 15: ATE_IND.1 Independent Testing – Conformance	
Developer Action Elements	
ATE_IND.1.1D	The developer shall provide the TOE for testing.
Content and Presentation Elements	
ATE_IND.1.1C	The TOE shall be suitable for testing.
Evaluator Action Elements	
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.

Table 16: AVA_VAN.1 Vulnerability Survey
Developer Action Elements

AVA_VAN.1.1D	The developer shall provide the TOE for testing.
Content and Presentation Elements	
AVA_VAN.1.1C	The TOE shall be suitable for testing.
Evaluator Action Elements	
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.

6.2.2 Security Assurance Requirements for the TOE

This ST conforms to the [NDcPP], which draws from the CC Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

6.3 Rationale

This ST claims exact conformance to cPP. Therefore:

- All secure usage assumptions, organizational security policies, and threats are completely covered by security objectives.
- Each objective counters or addresses at least one assumption, organizational security policy, or threat.
- The set of components (requirements) in the ST are internally consistent and complete.

7. TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following Security Functions:

- Security Audit

- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

7.1 Security Audit (FAU)

FAU_GEN.1, FAU_GEN.2

The TSF generates audit records of security relevant events using Linux rsyslog daemon. Individual processes write messages to the rsyslog daemon interface as soon as their events happen. The messages describe the events and their severity. When the AAA or CLI modules write their messages, they add the subject identity, i.e. username, to the messages whenever applicable. The rsyslog daemon adds time stamps to the messages from the system clock and routes them as soon as it receives them. There are two targets specified for rsyslog daemon to route messages to: local Flash and connection endpoint to the remote audit server. Thus, the rsyslog daemon makes one copy of audit log on the local Flash memory and sends another copy to the configured remote audit server as soon as they are generated Each audit record generated by the TSF includes the date and time stamp of when the event that generated the audit record occurred, type, subject identity (IP address, hostname, and/or username), the outcome (success or failure), and any additional information specified in Table 7.

The following security relevant events are logged.

- Start-up and shut-down of the audit functions:
  - The auditing functions can be stopped with the “no logging on” CLI command and started with the “logging on” CLI command.
- All of the following administrative action events:
  - Administrative login and logout
    - The name of the user account is included in the audit records.
  - Security related configuration changes
    - In addition to the information that a change occurred, what has been changed is included in the audit records.
  - Changes to cryptographic keys performed by security administrators:
    - Generating SSH key pair. The same key pair is used by both SSH server and client.
    - Generating key pair for Certificate Signing Request (CSR).
    - Importing of new TLS server certificate.

    In each of the above cases, SHA-256 hash of the public key or the certificate file is included in the audit log to uniquely identify them.
  - Resetting passwords

- These audit records include the name of related user account.

- Specifically defined auditable events listed in Table 7.

FAU_STG_EXT.1

Audit logs generated by the TSF are stored locally in the persistent Flash memory. The maximum size for the file storing the local audit logs is configurable. When the file exceeds its size limit, it is trimmed to remove the oldest audit logs until the size drops below the configured threshold. Locally stored audit records are protected from unauthorized viewing, modification and deletion by the file system's read/write permissions and a restrictive CLI which only allows identified, authorized and authenticated administrative users read/write access. The Security Administrative user can delete the locally stored audit records. Modification of the audit records other than deleting them is not supported.

The logs can also be sent to configured remote audit server in syslog format as soon as they are generated. To protect the audit records in transit from the TOE to the remote audit server in the Operational Environment, the TOE establishes a Trusted Channel between itself and the external audit server using the SSHv2 protocol. The Trusted Channel is created when the TOE establishes an SSH session between itself and the remote audit server with TCP port forwarding enabled. After the SSH session is established, the TOE is configured by the Security Administrative user to forward all messages received by the syslog process to the listening TCP port created by the SSH connection. This ensures that all audit traffic is encapsulated and hence protected by the SSH connection.

7.2 Cryptographic Support (FCS)

The EOS 4.22.1FX-CC contains Arista EOS Crypto Module v1.0 (CMVP certificate #2909) which uses underlying Fedora OpenSSL library for all cryptographic operations. The TSFs utilize CAVP validated cryptographic algorithms listed in Table 17.

Table 17: Cryptographic Algorithms

SFR	Description	Details	Cert. #
FCS_CKM.1 Cryptographic Key Generation	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	Key Generation: Public Key Exponent: Random Probable Primes with Conditions: Mod lengths: 2048 or 3072 (bits) Primality Tests: C.2 or C.3	RSA: 2301
FCS_CKM.2 Cryptographic Key Establishment	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1:		CVL: 1012

	RSA Cryptography Specifications Version 2.1"; Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;		
FCS_COP.1/Data En cryption	AES-CBC, AES-GCM as defined in NIST SP 800-38A	AES, Mode: CBC, Direction: Decrypt and Encrypt, Key Length: 128, 256	AES:4280
FCS_COP.1/SigG en	RSA schemes using cryptographic key sizes [of 2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4	Signature Generation PKCS1.5: Mod 2048 SHA: SHA-1 or SHA-256 or SHA-384 or SHA-512	RSA: 2301
FCS_COP.1/Hash	SHS that meets: FIPS Pub 180-4 or ISO/IEC 10118-3:2004. SHA Bit-oriented Mode Byte-oriented Mode	SHA-1 or SHA-256 or SHA-384 or SHA-512	SHS: 3516
FCS_COP.1/Keye dH ash	HMAC that meets : FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, "Secure Hash Standard or ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	HMAC-SHA2-256 Key Sizes < Block Size Key Sizes > Block Size Key Sizes = Block Size	HMAC: 2816
FCS_RBG_EXT.1 Random Bit Generation	CTR_DRBG(AES)	Counter DRBG, Mode: AES-256	DRBG: 1340

FCS_CKM.1

The TSF supports generation of 2048-bit RSA asymmetric keys for eAPI TLS server authentication and for SSH client and SSH server authentication. The supported RSA scheme meets the FIPS PUB 186-4, Digital Signature Standard (DSS), Appendix B.3 standard. The TSF generates Diffie-Hellman Group 14 asymmetric keys for session keys establishment in SSH and TLS session according to RFC 3526, Section 3.

FCS_CKM.2

Session keys for both SSH and TLS are generated with Diffie Hellman (DH) key exchange. OpenSSL library is used to perform DH operations of key pair generation and common secret computation using values of DH prime and DH generator are as specified in DH Group ID 14 standardized by IANA and provided in RFC 3526, Section 3.

FCS_CKM.4

The TOE destroys Critical Security Parameters (CSPs) when no longer required. Zeroization procedures for different CSPs in the TOE are described in Table 18.

Private keys of RSA used during SSH and TLS authentication are persisted in flash memory. If a persistent key is removed or replaced by the Security Administrator, the old persistent key is zeroized by a single direct overwrite consisting of zeroes.

The programmatic destruction of keys is carried out by using a specialized algorithm that overwrites the file location with successive random and all-zero patterns and then ensures that the key is destroyed by reading it back. This is done before writing the new key to the file. This ensures that any updates to the cryptographic key files (creation, modification, deletion) result in all previous key files being destroyed prior to the new values being written. As such, there is no delay in the destruction of keys. At the physical layer, non-volatile memory is implemented on the TOE by either eUSB, eMMC, or SSD devices. These have write endurance of at least 17 TB, which under normal usage are expected to last the lifetime of the device. In the unlikely event that there is a write-failure the issue will be reported in the logging system. In such circumstances it is recommended to replace the media and physically destroy the faulty non-volatile memory device.

DH keys and session keys of TLS and SSH are ephemeral and stored in RAM. They are zeroized by a single direct overwrite consisting of zeroes, by the time the corresponding session terminates.

Table 18: CSPs

CSP	Purpose	Generation	Clearing Method	When Cleared	Storage Location
TLS Server: RSA Private Key	"eAPI" server authentication	Generated internally when TLS server is first started or when new key pair is requested by	Single direct overwrite consisting of zeroes	When replaced by the Security Administrator	Flash
		Security Administrator			

TLS Server: DH Private Key	Establish session keys for TLS session	Generated internally at the start of TLS session	Single direct overwrite consisting of zeroes	When TLS session keys are derived	RAM
TLS Server: Session Keys	Message authentication and encryption in TLS session	Generated internally at the start of TLS session	Single direct overwrite consisting of zeroes	When TLS session terminates	RAM
SSH Server: RSA Private Key	SSH host authentication for remote administrative session	Generated internally when SSH service on the TOE is first started or when new key pair is requested by Security Administrator.	Single direct overwrite consisting of zeroes	When replaced by the Security Administrator	Flash
SSH Server: DH Private Key	Establish session keys for SSH session	Generated internally at the start of SSH session	Single direct overwrite consisting of zeroes	When SSH session keys are derived	RAM
SSH Server: Session Keys	Message authentication and encryption in SSH session	Generated internally at the start of SSH session	Single direct overwrite consisting of zeroes	When SSH session terminates	RAM
SSH Client: RSA Private Key	SSH client authentication to audit server	Generated internally when SSH service on the TOE is first started or when new key pair is requested by Security Administrator.	Single direct overwrite consisting of zeroes	When replaced by the Security Administrator	Flash
SSH Client: DH Private Key	Establish session keys for SSH session	Generated internally at the start of SSH session	Single direct overwrite consisting of zeroes	When SSH session keys are derived	RAM
SSH Client: Session Keys	Message authentication and encryption in SSH session	Generated internally at the start of SSH session	Single direct overwrite consisting of zeroes	When SSH session terminates	RAM

FCS_RBG_EXT.1 The TOE uses a software-based CTR_DRBG (AES-256) random bit generator (DRBG) that complies with NIST SP 800-90 for all cryptographic operations. Each DRBG instance is seeded with full 384 bits of entropy (256 bits for AES key and 128 bits for nonce) sourced from Linux Random Number Generator (LRNG) operating in a blocking mode (/dev/random). LRNG accumulates entropy from two software based noise sources: i) interrupt times via `add_interrupt_randomness` function, and ii) variability in timing of instructions executions in CPU using "Haveged" daemon. The detailed entropy justification is provided in [ENT].

FCS_SSHC_EXT.1, FCS_SSHS_EXT.1

TOE utilizes SSHv2 to support Trusted Channel to external audit server and Trusted Path for remote administration session. SSH implementation conforms to the following RFCs:

- 4251 - The Secure Shell (SSH) Protocol Architecture
- 4252 - The Secure Shell (SSH) Authentication Protocol
- 4253 - The Secure Shell (SSH) Transport Layer Protocol
- 4254 - The Secure Shell (SSH) Connection Protocol
- 6668 - SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol
- 8332 - Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol

TOE performs the role of SSH client in Trusted Channel. The TSF ensures that the SSH client authenticates the identity of the audit server using the /etc/ssh/known_hosts file which associates each server host name with its corresponding public key as described in RFC 4251. For this, the TSF compares the received host key from the audit server during the SSH handshake and compares it to the keys configured in the known_hosts file. If there is a match, the connection establishment process proceeds. If there is no match found, the session is terminated. SSH client authenticates to the audit server using public key.

TOE performs the role of SSH server in Trusted Path. SSH server in the TOE authenticates remote administrative user using public key or password. Following public key scheme is supported: rsa-sha2-256 that uses 2048-bit RSA key and SHA-256 digital signature. Additional details of remote administrative user authentication are provided in FIA_UIA_EXT.1.

SSH session keys are established using DH key exchange. The scheme supported is: diffie-hellman-group14-sha1. It supports 2048-bit asymmetric keys (DH Group 14). It uses SHA-1 for exchange hash. Exchange hash is the hashing method used to generate session keys hierarchy from the shared secret derived from DH key establishment.

Encryption/decryption cipher supported is AES-CBC with 128 and 256 key lengths. Message authentication code supported is hmac-sha2-256.

In order to comply with RFC 4253, "large packets" received by the SSH client (packets greater than 262,144-bytes) in the SSH connection are dropped.

The SSH performs rekeying when 1 GB of SSH data has been sent, or after 1 hour of that session being open. A counter in the SSH code keeps track of the sent SSH data packets. If the data counter reaches 1 GB, or if the time counter reaches 1 hour, the TSF sends a 'Key Exchange Init' message to the peer to initiate a new key-exchange negotiation. If the new key-exchange fails to establish a new key for the session, the session is terminated by the TSF.

FCS_TLSS_EXT.2

The TOE allows for automated remote management of the TSF via the eAPI JSON-RPC ("eAPI") interface.

The communication channel is protected by TLS TLSv1.2 with mutual authentication. Any attempts to establish a session using any other TLS or SSL versions (SSL 1.0, SSL, 2.0, SSL 3.0, TLS 1.0, TLS 1.1) is denied by the TSF. Following TLS ciphersuites are supported:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246

TOE acts as a TLS server to the eAPI TLS client in the operational environment. To support the TLS server authentication to the eAPI client, the TSF needs to be configured with an RSA 2048-bit public-key in the x.509v3 certificate format.

When the TSF is establishing a TLS session utilizing a ciphersuite with RSA key establishment (key transport), the TSF only acts as the receiver of the encrypted pre-master secret. When the TSF is establishing a TLS session utilizing a ciphersuite with DHE key establishment, the TSF generates DH Group 14 key pair and sends public key from the pair to the peer to facilitate pre-master secret establishment.

The TLS server supports authentication of the eAPI TLS client. The Security Administrator configures TLS server with trusted CA certificate used to validate the eAPI TLS client certificate. The trusted CA certificate is then associated to a Security Administrative user account. This particular administrative account is only to be used by the eAPI. This is to ensure that there is no ambiguity when viewing the audit records as to whether a human user's action or the eAPI's action generated the audit record. When eAPI TLS client attempts to establish a session with the TSF, the TSF responds to the client by sending a Certificate_Request message immediately after the ServerKeyExchange message is sent during the TLS handshake. The client must send a Client_Certificate message containing an RSA 2048-bit public-key in the x.509v3 certificate to authenticate the client to the TSF. The TSF validates the client certificate according to FIA_X509_EXT.1.1, checks that it is signed by the trusted CA, checks the local CRL for the revocation status of the client certificate. If any of these validity checks fail, the session is terminated.

If the client certificate validity checks passes, the TSF checks to see if the CN value matches the username of the account specifically configured for the eAPI client. If no match is found, TSF will terminate the attempted session establishment. If a match is found, TSF allows the authentication process to complete and the session to successfully establish. The TSF does not process SAN value in the client certificate.

7.3 Identification and Authentication (FIA)

FIA_AFL.1

Consecutive authentication failures result into temporary account lockout for remote administrative user. The threshold number of failures between 1 and 255 and subsequent lockout period are configured by Security Administrator when initializing the TOE. The RS-232/VT-100 local administrative interface is never locked out.

FIA_PMG_EXT.1

The Security Administrator passwords are able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".

Minimum length of the password is configurable by the Security Administrator between 1 and 32. It is recommended to configure minimum length as at least 8.

FIA_UIA_EXT.1, FIA_UAU_EXT.2

The TSF provides a local administrative CLI interface over RS-232/VT-100 that supports password-based authentication. It also provides remote administrative CLI interface over SSH that supports password-based and public key based authentication.

In password based authentication, once a user initiates a connection to the administration interface they are prompted to provide username and password credentials. After a user provides a username and password, the TOE invokes a PAM (Pluggable Authentication Modules) AAA plugin. This plugin is configured to use the RBAC (Role Based Access Control) module. The AAA plugin passes the authentication credentials to the RBAC module. The RBAC module checks the credentials against its database and then responds with a SUCCESS or DENIED message. Based on the response from the RBAC module, the AAA plugin then returns a SUCCESS or DENIED message to the requesting program. If the requesting program receives a DENIED message it prints an error message to the user and denies the login attempt. If the requesting program receives a SUCCESS message, it makes an EXEC authorization request to the RBAC module, which responds with the additional permissions for the CLI. At this point, authentication is successful and the user is presented with the CLI.

When RSA public key authentication is used with a remote SSH session, the SSH daemon performs the initial authentication verification locally by comparing public key supplied by the client with /home/<account>/.ssh/authorized_keys. Once the public key matches, the daemon performs an EXEC authorization request to the RBAC module as detailed above.

The TSF also provides “eAPI” interface over TLS for remote automated configuration. The details of eAPI authentication process are provided above in FCS_TLSS_EXT.2. After the authentication is successful, EXEC authorization request is made to the RBAC module.

The TSF displays a warning banner (in accordance with FTA_TAB.1) on CLI prior to requiring identification and authentication. This banner is not required to be, nor is it, presented to the eAPI JSON-RPC client prior to it authenticating to the TSF.

FIA_UAU.7

The TSF does not echo back the characters that are entered when users attempt to authenticate to the local console when using a password credential.

FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3

TOE performs the role of TLS server in eAPI trusted channel communication. Security Administrator installs x.509v3 certificate in the TLS server. To facilitate this, Security Administrator first generates certificate signing request (CSR). CSR causes generation of 2048-bit RSA private and public key pair in the TOE. It encapsulates the public key, along with Common Name, Organization, Organizational Unit, and Country information, in a format that can be submitted to certificate authority (CA) for creating signed x.509v3 certificate. Security Administrator obtains the signed certificate from the CA and imports in the TOE. Following conditions are checked at the time of import and the import is rejected if any of the following conditions do not check:

- That the entire chain of certificates is imported. That is, the leaf TLS server certificate, any intermediate certificates leading from the leaf up to the root and the root certificate are imported.

- That the current date and time lies between the “Valid from” and “Valid to” for each certificate.
- That the basicConstraints extension is included with CA flag is set to TRUE for all CA certificates in the chain
- That the extendedKeyUsage field in the leaf certificate has the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1)
- That the digital signatures are correct in all certificates
- That none of the certificates from the leaf up to the root is revoked.

The above certificate chain is presented to the eAPI client at the beginning of TLS connection. This facilitates the eAPI client to validate identity of the server.

At the time of establishing TLS connection, the eAPI client presents its x.509v3 certificate to the TLS server in the TOE. This certificate is used by the TOE to authenticate the eAPI client. The TOE performs following checks on the certificate presented by the eAPI client:

- That the certificate chain presented by the eAPI client can be traced to the CA certificate that is lowest in the hierarchy of certificates imported into the TOE as described above. That is, this lowest CA certificate acts as the trust anchor. Note that the trust anchor can be an intermediate CA certificate or the root CA certificate.
- That the current date and time lies between the “Valid from” and “Valid to” for each certificate from the leaf certificate in the chain presented by eAPI client upto and including the trust anchor.
- That the basicConstraints extension is included with CA flag is set to TRUE for all CA certificates in the chain.
- That the extendedKeyUsage field in the leaf certificate in the chain presented by eAPI client has the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2).
- That the digital signatures are correct in all certificates.
- That none of the certificates from the leaf up to the trust anchor is revoked. It is not necessary to check revocation status of the trust anchor and other CA certificates upstream of the trust anchor.

If any of the checks mentioned above fails, then the TLS connection is rejected.

In order to facilitate revocation checking on the eAPI client certificate chain as described above, Security Administrator specifies Certificate Distribution Points (CDPs) during initial configuration of the TOE. Security Administrator is required to specify CDPs for the CRLs published by the trust anchor and every CA certificate between the trust anchor and the leaf certificate of the eAPI client. The TOE downloads CRLs from the specified CDPs every 24 hours and stores the most recently fetched CRLs locally. Digital signatures on downloaded CRLs are validated and in case of failure of signature verification, CRL is not added to the local copy. The local copies of the CRLs are used for certificate revocation checking. At the time of revocation checking, the TOE ensures that the current time lies within the validity period of the CRL, that is between the effective date and the next update date mentioned in the CRL. The certification revocation checking can fail either because the certificate is revoked as per the CRL or because the recent CRL for the CA that issued the certificate is not present in the local copy to perform the revocation checking.

7.4 Security Management (FMT)

FMT_MOF.1

See description under **FPT_TUD_EXT.1.**

FMT_MTD.1, FMT_SMF.1, FMT_SMR.2

The TSF maintains the 'Security Administrator' role and allows that role to be associated to users of the TOE. The Security Administrator role is enforced via the permission features of the base Linux kernel of the EOS operating system. Users are associated with credentials for identification and role for authorization to the TOE. In addition, users' permissions are assigned and enforced by the base Linux kernel filesystem of EOS via read/write/execute permissions and group policies that ensure that only Security Administrative users can manage the TSF data. The TSF restricts the ability to manage (i.e. create, view, initialize, modify/append, delete/clear, and change the default setting) of the TSF data to only identified, authenticated and authorized Security Administrators.

The local console and remote management interfaces allow the Security Administrator to perform the following TSF management functions:

- Administer the TOE locally and remotely
- Create the TOE access banner
- Set the session inactivity timeout values
- Verify and manually install firmware updates (verification using published hash)
- Configure failed login threshold and lockout period
- Generate, import, delete and configure cryptographic keys required by SSH and TLS
- Specify ciphersuites for SSH and TLS
- Set system time
- Import x.509v3 certificates in trust store

7.5 Protection of the TSF (FPT)

7.5.1 Protection of Administrator Passwords

FPT_SKP_EXT.1

Refer to Tables 18 for a list of private keys and symmetric keys. TOE stores these keys in plaintext is locations described for them in Table 18. Administrative interface does not provide a documented command for any user to view any of the private or session keys. There are no pre-shared keys.

FPT_APW_EXT.1

The TSF does not store passwords in plaintext. The TSF uses SHA-512 hash (with a salt) protection to ensure that any users' password is not stored in plaintext. When a user authenticates to the TSF (local or remote), the system generates a hash of the entered password and compares it against the stored hash value of the password associated to the username provided to ensure that the plaintext password is not disclosed.

FPT_TST_EXT.1

The first self-test checks the software integrity of the cryptographic library as a whole by calculating the HMAC-SHA256 value of the core binaries and comparing it against the value calculated at compile-time.

The cryptographic self-tests are run whenever programs which utilize the cryptographic libraries are loaded. This includes the instances when the SSH server is started, when the SSH client is started, when

the TLS server is started and when 'FIPS mode' is enabled on the TOE. The cryptographic self-tests are test each cryptographic algorithms utilized by SSH and TLS implementations in the TSF. They include Known Answer Tests (KAT) for RSA, AES, HMAC, SHA and DRBG.

Pairwise Consistency Test is run upon generation of RSA key pair. If any of these tests fail, the TSF services that rely on the cryptographic functionality that failed testing would become unavailable and audit record would be generated. Continuous test (CRNGT) for stuck fault is performed on DRBG.

FPT_TUD_EXT.1

Software updates are verified using SHA-512 published hash values. The update is performed by Security Administrator and only on the SSH remote management console. The candidate update package is downloaded from the Arista's customer portal. Arista customers require a valid credential to login to this portal to obtain the candidate updates. The candidate updates are provided with a separate download of the SHA-512 hashed value of the update package. The Security Administrator is required to transfer the downloaded candidate package from a trusted terminal to the TOE using secure means; typically SCP (secure copy) is utilized. The SHA-512 checksum of the candidate file is to be generated on the candidate update package that was securely copied to the TOE. This is done by issuing the following command: *sha512sum <update-package>*. The Security Administrator must verify that the on-screen output of the checksum of the candidate update package matches the published hash of the candidate update package before initiating the installation of the candidate update package. The Security Administrator accomplishes this by visually comparing the published checksum to the checksum of the candidate update package that was generated by the Security Administrator. If the hash generated does not match the published hash for the candidate update, the Security Administrator is instructed not to proceed further.

Security Administrator can verify the version of currently running EOS with "show version" CLI command.

FPT_STM_EXT.1

The following TSFs make use of the system time:

- Time stamping of TSF generated audit records
- Refetching of the new CRL at the end of validity period of the previous CRL and to check that the time when the certificate is presented to the TOE lies within the validity period of the CRL
- Time keeping of interactive sessions between the Security Administrator and the TSF (local console, remote SSH console interfaces) for the purpose of TSF termination of the interactive sessions due to inactivity.

The initial system time is manually configured by the Security Administrator using the CLI interface. After that, two components are responsible to keep the system time, namely, the Real Time Clock (RTC) and the "system clock". The RTC is battery powered and keeps track of time even when the TOE is not provided with power from an A/C source. The system clock is a software counter based on the timer interrupt. The system clock is reset after power cycle, and is initialized by the RTC during boot-up. The RTC is maintained by an oscillator with an accuracy of 20 PPM.

The system clock and RTC can be expected to drift over time since both lack access to a reference source. Over the course of one month, this drift will be linear and has small upper bounds. For the RTC the absolute value of the drift is expected to be 1 minute or less per 30 days. For the system clock the absolute value of the drift is expected to be 3.9 seconds or less per 30 days. In order to limit the amount of drift, Security Administrator will be expected to set the clock on the TOE using a CLI command, once every 30

days. This new time must come from legitimate time source, meeting Common Criteria requirements. Upon setting the time via the CLI command, both the system clock and the RTC will be set and the amount of drift from the real time will be reduced to 0.

7.6 TOE Access (FTA)

FTA_SSL_EXT.1, FTA_SSL.3

Security Administrator can configure a time period to be used to automatically terminate administrative session after inactivity. For local and remote administrative session, the TSF terminates the session after this period of inactivity. To facilitate this, a timer is started for the session after successful authentication of Security Administrator. The timer is reset each time the Security Administrator provides input. If the Security Administrator does not provide input for a duration of configured inactivity period, the TSF terminates the administrative session.

FTA_SSL.4

The TSF provides the means for the Security Administrator to manually terminate their own interactive sessions with the TOE for both the local and remote administrative sessions. For this, the Security Administrator can issue the “quit” or “exit” command to terminate their own session.

FTA_TAB.1

Before establishing an administrative user session to the TSF, the TSF displays a Security Administrator-specified advisory notice and consent warning message (banner). The banner is presented prior to human user authentication on each of the TOE’s administrative interfaces. This banner is not required to be, nor is it, presented to the eAPI JSON-RPC client prior to it authenticating to the TSF. The banner can be configured via any one of the following administrative interfaces: Local console (serial port), remote console (SSH), remote management (eAPI JSON-RPC client).

7.7 Trusted Path/Channels (FTP)

FTP_ITC.1

Trusted Channel connection is created between TSF and audit server. It is protected by SSH. TOE acts as SSH Client in the Trusted Channel connection to audit server. Operation of SSH is described above in FCS_SSHC_EXT.1.

Trusted Channel connection is created between TSF and eAPI JSON-RPC Client. It is protected by TLS. TOE acts as TLS Server in the Trusted Channel connection to eAPI JSON-RPC Client. Operation of TLS is described above in FCS_TLSS_EXT.2.

FTP_TRP.1

Trusted Path connection protects communication between TSF and human user (Security Administrator) performing management of the TSF. It is protected by SSH. TOE acts as SSH Server in the Trusted Path connection. Operation of SSH is described above in FCS_SSHS_EXT.1.

8. Terms and Definitions

Table 19: TOE Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
AAA	Authentication Authorization and Accounting
CLI	Command Line Interface
CPU	Central Processing Unit
CSP	Critical Security Parameter
JSON	JavaScript Object Notation
KAT	Known Answer Tests
OSI	Open Systems Interconnection
PAM	Pluggable Authentication Modules
RBAC	Role Based Access Control
RPC	Remote Procedure Call
SSH	Secure Shell
TLS	Transport Layer Security

Table 20: CC Abbreviations and Acronyms

Abbreviations / Acronyms	Description
CC	Common Criteria
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

9. References

Table 21: TOE Guidance Documentation

Reference	Description	Date
[T1]	User Manual - Arista EOS version 4.21.0F	August 06, 2018
[T1]	Common Criteria Guidance Supplement	October 17, 2019

Table 22: Common Criteria v3.1 References

Reference	Description	Version	Date

[C1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model CCMB-2009-07-001	V3.1 R5	April 2017
[C2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components CCMB-2009-07-002	V3.1 R5	April 2017
[C3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components CCMB-2009-07-003	V3.1 R5	April 2017
[C4]	Common Criteria for Information Technology Security Evaluation. Evaluation Methodology CCMB-2009-07-004	V3.1 R5	April 2017

Table 23: Supporting Documentation

Reference	Description	Version	Date
[S1]	Collaborative Protection Profile for Network Devices	2.1	September 14, 2018
[ENT]	Entropy Analysis Report - Arista Data Center Switches Running EOS	1.1	December 17, 2018