# Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform Security Target

Version 1.1
02/17/2025

*Prepared for:*

ciena

7035 Ridge Rd,
Hanover, MD 21076

*Prepared By:*

Gossamer Laboratories

www.gossamersec.com

## LIST OF TABLES

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Ciena SAOS R10.9.1 on 3926 with Large NFV Compute Server Service Aggregation Platform provided by Ciena Corporation. The TOE is being evaluated as a network device with MACsec.

The Security Target contains the following additional sections:

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

***Conventions***
The following conventions have been applied in this document:

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

## 1.1 Security Target Reference

**ST Title –** Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform Security Target

**ST Version** – Version 1.1

**ST Date** –02/17/2025

## 1.2 TOE Reference

**TOE Identification** – Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform

**TOE Developer** – Ciena Corporation

**Evaluation Sponsor** – Ciena Corporation

## 1.3 TOE Overview

The Target of Evaluation (TOE) is the Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform. The TOE is a non-distributed, non-virtual network device which implements routing and switching functionalities for enterprise, mobility, and converged network architectures. In these architectures, the TOE can be deployed in the access, aggregation, or core of the network. The TOE uses a Linux based container architecture for its SAOS Network Operating System and includes the Ciena SAOS R10.9.1 operating system executed on the 3926 with Large NFV Compute Server Service Aggregation Platform.

## 1.4 TOE Description

The TOE implements the general functionality of a router/switch consistent with the collaborative Protection Profile for Network Devices v2.2E with MACsec module v1.0. The TOE implements controlled connectivity between two subnetworks and a management interface. All network traffic between the connected subnetworks is controlled by the TOE and the authorized administrators may manage the TOE using the management interface.

The management interface is a Command Line Interface (CLI) which may be accessed locally or remotely. Local access is via a console port which is a Serial EIA-561 (RJ-45) or a USB-C port. It allows management of the TOE from a workstation physically connected to the TOE. Remote management is over Secure Shell (SSH). SSH implements a secure remote login over a network connection and allows protected CLI and Network Configuration Protocol (NETCONF) access to the TOE.

All administrators are identified and authenticated using a username and password or based on SSH public key authentication. Access is only granted, and the user assigned to the role administrator upon successful authentication. Authentication is implemented locally. Authentication of TLS peers is done using X.509 Public Key Certificates. The validity of the X.509 public key certificates is verified using the Online Certificate Status Protocol (OCSP). TLS may also be used for forwarding audit records to an external audit server. MACsec is used for protected communication among MACsec peers.

### 1.4.1 TOE Architecture

The TOE is the Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform. The platform is a single instance with the following physical characteristics:

- 3926-905
  - Processor - ARM Cortex A53, 4CORE
  - ASIC –Broadcom BCM82759 MACSec
- Large NFV Compute Server
  - Processor - Intel XEON D1548, 8CORE

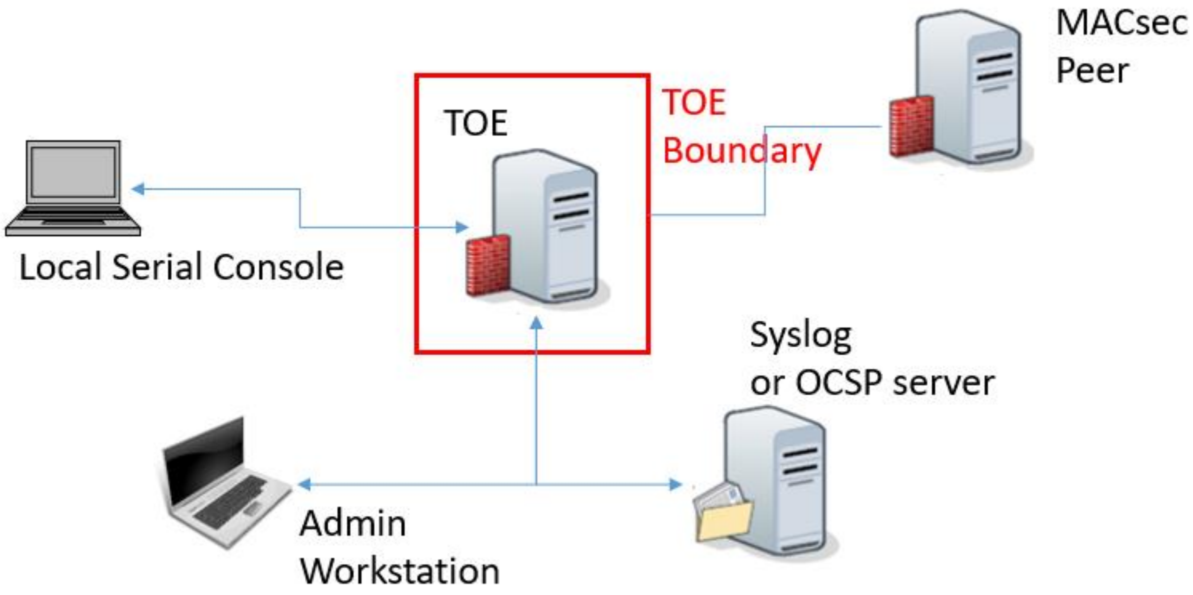
#### 1.4.1.1 Physical Boundaries

The TOE is composed of the following components:

- TOE software is Ciena SAOS R10.9.1
- TOE hardware is the Ciena 3926-905 with Large NFV Compute Server Service Aggregation Platform

The TOE is deployed in an environment that includes the IT components illustrated in the following figure. The TOE itself is delivered as an appliance or an FRU with the software installed. The administrator of the TOE may verify the TOE software and, if necessary, download and install the correct version.

The physical boundary of the TOE is illustrated below. Non-TOE components are summarized in **Table 1 IT Environment Components**. The TOE implements a TLS Client and SSH Server for secure connectivity to the components of the environment. Each component of the environment is required to implement the corresponding client and/or server. The remote management workstation is required to implement a SSH Client for accessing the TOE.



The environmental components described in the following table are required to operate the TOE in the evaluated configuration.

| Component | Description |
|---|---|
| MACsec Peer | A peer server that supports MACsec to communicate with the TOE. |
| Audit (syslog) server (Mandatory) | The audit server supports syslog messages over TLS v1.2 to receive the audit files from the TOE. The audit data is stored in the remote audit server for redundancy purposes. |
| OCSP Server (Mandatory) | Validity of the certificates the TOE uses for asserting the authenticity of the TLS peers is verified using OCSP. |
| Admin Workstation (Mandatory) | A workstation used by an administrator to manage the TOE locally or remotely. The remote management station must include a SSH client. |

**Table 1 IT Environment Components**

#### 1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

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

#### 1.4.1.2.1 Security audit

The TOE is able to generate logs for a wide range of security relevant events. The TOE can be configured to store the logs locally so they can be accessed by an administrator and also to send the logs to a designated log server using TLS to protect the logs while in transit on the network. Each audit record contains the date and time of event, type of event, subject identity, and any other event-related relevant data

#### 1.4.1.2.2 Cryptographic support

The TOE contains a CAVP-tested cryptographic module that provides key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher-level cryptographic protocols including MACsec, SSH and TLS.

#### 1.4.1.2.3 Identification and authentication

The TOE authenticates all users connecting to the management interfaces of the TOE. Authentication may take place over the console for local access or over SSH for remote access. Only upon successful authentication, given that the user is authorized for the role, is a user assigned to the role administrator and granted access to the management functions of the TOE. Local users authenticate to the TOE using a username and password. Remote users may also use SSH public-key authentication. A customizable warning banner is displayed at each authentication window.

Users may change their own passwords, but the TOE enforces minimum quality criteria for the passwords. The TOE also maintains a counter for consecutive, failed authentication attempts for each user. If the counter value reaches an administrator-defined threshold, the TOE triggers protective measures to prevent password guessing attacks.

The TOE uses X.509v3 certificates for peer entity authentication of TLS peers. The TSF determines the validity of the certificates by confirming the validity of the certificate chain and verifying that the certificate chain ends in a trusted Certificate Authority (CA). The TOE connects to an Online Certificate Status Protocol (OCSP) server to confirm the revocation status of the certificates.

#### 1.4.1.2.4 Security management

The TOE allows local and remote management of its security functions. Local management is from a management workstation connected to the console or USB port of the TOE and the remote management is from a workstation connected to the TOE over SSHv2.

All management functions are implemented using the CLI and are only made available to authorized administrators upon successful identification and authentication. The TOE also supports a NETCONF interface which is also protected with SSH.  The administrator can use the NETCONF interface to perform the same functions as the CLI.

#### 1.4.1.2.5 Protection of the TSF

The TOE protects all passwords, pre-shared keys, symmetric keys, and private keys from unauthorized disclosure. Pre-shared keys, symmetric keys, and private keys are stored encrypted. An exception is the SSH host keys which are required by the SSH Daemon when the SSH starts. The SSH Daemon is executed at the root privileges and the SSH host keys are only accessible with root privileges. No user of the TOE is granted root privileges. Passwords are stored as non-reversible hash values computed using SHA-512 using the shadow utility functions. The TOE maintains system time via its local hardware clock which is manually set by an administrator.

The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. An administrator can install software updates to the TOE and verify the authenticity of all updates prior to the installation.

#### 1.4.1.2.6 TOE access

Prior to establishing an administration session with the TOE, an access banner is displayed to the user. The banner messaging is customizable but is typically used to warn the users of the consequences of attempted unauthorized access. The TOE will terminate an interactive session after a configurable time of session inactivity. A user may terminate his/her local and remote administrative sessions on will.

#### 1.4.1.2.7 Trusted path/channels

The TOE implements trusted paths and trusted channels. The trusted path is a SSH connection between the TOE and the remote management workstation. The SSH client of the remote management workstation connects to the SSH Server implemented by the TOE. Upon successful connection establishment and authentication of the user, the remote administrator uses the CLI or NETCONF over SSH to manage the TOE.

For trusted channels, the TOE implements a TLS client used by the TOE to connect to the peer entities. The peer entity is a remote syslog server. The TOE uses MACsec to protect communications with MACsec peers.

### 1.4.2 TOE Documentation

Ciena SAOS R10.9.1 on 3926 with Large NFV Compute Server Service Aggregation Platform CC Guidance Supplement v1.0

# 2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - PP-Configuration for Network Devices and MACsec Ethernet Encryption, Version 1.0, 2023-03-29 (CFG_NDcPP-MACsec_V1.0)
        - Base PP: collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 (NDcPP22e)
        - PP Module: PP-Module for MACsec Ethernet Encryption, Version 1.0, 02 March 2023 (MACSEC10)

| Package | Technical Decision | Applied | Notes |
|---|---|---|---|
| NDcPP22e | TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance | No | Requirement not claimed |
| NDcPP22e | TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR | Yes | |
| NDcPP22e | TD0790: NIT Technical Decision: Clarification Required for testing IPv6 | Yes | |
| NDcPP22e | TD0738: NIT Technical Decision for Link to Allowed-With List | Yes | |
| NDcPP22e | TD0670 - NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing | Yes | |
| NDcPP22e | TD0639 - NIT Technical Decision for Clarification for NTP MAC Keys | No | Requirement not claimed |
| NDcPP22e | TD0638 - NIT Technical Decision for Key Pair Generation for Authentication | Yes | |
| NDcPP22e | TD0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH | No | Requirement not claimed |
| NDcPP22e | TD0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters | No | Requirement not claimed |
| NDcPP22e | TD0632 - NIT Technical Decision for Consistency with Time Data for vNDs | Yes | |
| NDcPP22e | TD0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server | Yes | |
| NDcPP22e | TD0592 - NIT Technical Decision for Local Storage of Audit Records | Yes | |
| NDcPP22e | TD0591 - NIT Technical Decision for Virtual TOEs and hypervisors | Yes | |
| NDcPP22e | TD0581 - NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3 | Yes | |
| NDcPP22e | TD0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e | Yes | |

| | | | |
|---|---|---|---|
| NDcPP22e | TD0572 - NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers | Yes | |
| NDcPP22e | TD0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1 | Yes | |
| NDcPP22e | TD0570 - NiT Technical Decision for Clarification about FIA_AFL.1 | Yes | |
| NDcPP22e | TD0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 | No | Requirement not claimed |
| NDcPP22e | TD0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria | Yes | |
| NDcPP22e | TD0563 - NiT Technical Decision for Clarification of audit date information | Yes | |
| NDcPP22e | TD0556 - NIT Technical Decision for RFC 5077 question | No | Requirement not claimed |
| NDcPP22e | TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test | No | Requirement not claimed |
| NDcPP22e | TD0547 - NIT Technical Decision for Clarification on developer disclosure of AVA_VAN | Yes | |
| NDcPP22e | TD0546 - NIT Technical Decision for DTLS - clarification of Application Note 63 | No | Requirement not claimed |
| NDcPP22e | TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 | Yes | |
| NDcPP22e | TD0536 - NIT Technical Decision for Update Verification Inconsistency | Yes | |
| NDcPP22e | TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4 | No | Requirement not claimed |
| NDcPP22e | TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1) | Yes | |
| MACSEC10 | TD0891 - Correlation of Implicitly Satisfied Requirements when CPP_ND_V3.0E is the Base-PP | Yes | |
| MACSEC10 | TD0889 - Correction For Tests Incorrectly Requiring Group MACsec | Yes | |
| MACSEC10 | TD0884 - Expansion of Permitted EtherTypes in FCS_MACSEC_EXT.1.4 | Yes | |
| MACSEC10 | TD0882 - MACsec Data Delay Protection, Key Agreement, and Conditional Support for Group CAK | Yes | |
| MACSEC10 | TD0881 - Correction to MN Usage for FPT_RPL.1 Test | Yes | |
| MACSEC10 | TD0870 - Security Objectives Rationale for MOD_MACSEC_V1.0 | Yes | |
| MACSEC10 | TD0840 - Alignment of Test 22.1 to FMT_SMF.1/MACSEC | Yes | |
| MACSEC10 | TD0826 - Aligning MOD_MACSEC_V1.0 with CPP_ND_V3.0E | No | Not claiming CPP_ND_V3.0E |
| MACSEC10 | TD0816 – Clarity for MACsec Self Test Failure Response | Yes | |
| MACSEC10 | TD0803 – Clarification for Configurable MACsec CKN Length | Yes | |
| MACSEC10 | TD0746 – Correction to FPT_RPL.1 Test 25 | Yes | |
| MACSEC10 | TD0728 – Corrections to MACSec PP-Module SD | Yes | |

## 2.1 Conformance Rationale

The ST conforms to the NDcPP22e/MACSEC10. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

# 3. Security Objectives

The Security Problem Definition may be found in the NDcPP22e/MACSEC10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP22e/MACSEC10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP22e/MACSEC10 should be consulted if there is interest in that material.

In general, the NDcPP22e/MACSEC10 has defined Security Objectives appropriate for network device with MACsec and as such are applicable to the Ciena SAOS R10.9.1 on the 3926 with Large NFV Compute Server Service Aggregation Platform TOE.

## 3.1 Security Objectives for the Operational Environment

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

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

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

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

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

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

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

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

**OE.VM_CONFIGURATION** (applies to vNDs only)

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

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

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

# 4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP22e/MACSEC10. The NDcPP22e/MACSEC10 defines the following extended requirements and since they are not redefined in this ST the NDcPP22e/MACSEC10 should be consulted for more information in regard to those CC extensions.

**Extended SFRs:**

- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- MACSEC10:FCS_MACSEC_EXT.1: MACsec
- MACSEC10:FCS_MACSEC_EXT.2: MACsec Integrity and Confidentiality
- MACSEC10:FCS_MACSEC_EXT.3: MACsec Randomness
- MACSEC10:FCS_MACSEC_EXT.4: MACsec Key Usage
- MACSEC10:FCS_MKA_EXT.1: MACsec Key Agreement
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631
- NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0670 & TD0790
- NDcPP22e:FIA_PMG_EXT.1: Password Management - per TD0792
- MACSEC10:FIA_PSK_EXT.1: Pre-Shared Key Composition
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
- MACSEC10:FPT_CAK_EXT.1: Protection of CAK Data
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
- NDcPP22e:FPT_TST_EXT.1: TSF testing
- NDcPP22e:FPT_TUD_EXT.1: Trusted update
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking

# 5. Security Requirements

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

The SFRs have all been drawn from the NDcPP22e/MACSEC10. The refinements and operations already performed in the NDcPP22e/MACSEC10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP22e/MACSEC10 and any residual operations have been completed herein. Of particular note, the NDcPP22e/MACSEC10 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP22e/MACSEC10. The NDcPP22e/MACSEC10 should be consulted for the assurance activity definitions.

## 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Ciena SAOS R10.9.1 with MACsec on the 3926 Service Aggregation Platform TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security audit** | NDcPP22e:FAU_GEN.1: Audit Data Generation |
| | MACSEC10:FAU_GEN.1/MACSEC: Audit Data Generation (MACsec) |
| | NDcPP22e:FAU_GEN.2: User identity association |
| | NDcPP22e:FAU_STG.1: Protected audit trail storage |
| | NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage |
| **FCS: Cryptographic support** | NDcPP22e:FCS_CKM.1: Cryptographic Key Generation |
| | NDcPP22e:FCS_CKM.2: Cryptographic Key Establishment |
| | NDcPP22e:FCS_CKM.4: Cryptographic Key Destruction |
| | MACSEC10:FCS_COP.1/CMAC: Cryptographic Operation (AES-CMAC Keyed Hash Algorithm) |
| | NDcPP22e:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption) |
| | NDcPP22e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm) |
| | NDcPP22e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm) |
| | MACSEC10:FCS_COP.1/MACSEC: Cryptographic Operation (MACsec AES Data Encryption and Decryption) |
| | NDcPP22e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification) |
| | MACSEC10:FCS_MACSEC_EXT.1: MACsec |
| | MACSEC10:FCS_MACSEC_EXT.2: MACsec Integrity and Confidentiality |
| | MACSEC10:FCS_MACSEC_EXT.3: MACsec Randomness |
| | MACSEC10:FCS_MACSEC_EXT.4: MACsec Key Usage |
| | MACSEC10:FCS_MKA_EXT.1: MACsec Key Agreement |
| | NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation |
| | NDcPP22e:FCS_SSHS_EXT.1: SSH Server Protocol - per TD0631 |
| | NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0670 & TD0790 |

| | |
|---|---|
| **FIA: Identification and authentication** | NDcPP22e:FIA_AFL.1: Authentication Failure Management |
| | NDcPP22e:FIA_PMG_EXT.1: Password Management - per TD0792 |
| | MACSEC10:FIA_PSK_EXT.1: Pre-Shared Key Composition |
| | NDcPP22e:FIA_UAU.7: Protected Authentication Feedback |
| | NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism |
| | NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication |
| | NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation |
| | NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication |
| **FMT: Security management** | NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour |
| | NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data |
| | NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data |
| | NDcPP22e:FMT_SMF.1: Specification of Management Functions - per TD0631 |
| | MACSEC10:FMT_SMF.1/MACSEC: Specification of Management Functions (MACsec) - per TD0748 |
| | NDcPP22e:FMT_SMR.2: Restrictions on Security Roles |
| **FPT: Protection of the TSF** | NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords |
| | MACSEC10:FPT_CAK_EXT.1: Protection of CAK Data |
| | MACSEC10:FPT_FLS.1: Failure with Preservation of Secure State |
| | MACSEC10:FPT_RPL.1: Replay Detection - per TD0746 |
| | NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) |
| | NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632 |
| | NDcPP22e:FPT_TST_EXT.1: TSF testing |
| | NDcPP22e:FPT_TUD_EXT.1: Trusted update |
| **FTA: TOE access** | NDcPP22e:FTA_SSL.3: TSF-initiated Termination |
| | NDcPP22e:FTA_SSL.4: User-initiated Termination |
| | NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking |
| | NDcPP22e:FTA_TAB.1: Default TOE Access Banners |
| **FTP: Trusted path/channels** | NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel - per TD0639 |
| | MACSEC10:FTP_ITC.1/MACSEC: Inter-TSF Trusted Channel (MACsec Communications) |
| | NDcPP22e:FTP_TRP.1/Admin: Trusted Path - per TD0639 |

**Table 2 TOE Security Functional Components**

## 5.1.1 Security audit (FAU)

### 5.1.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1)

**NDcPP22e:FAU_GEN.1.1**

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).

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

d) Specifically defined auditable events listed in **Table 3**.

**NDcPP22e:FAU_GEN.1.2**

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of **Table 3**.

| **Requirement** | **Audit Event** | **Additional Contents** |
|---|---|---|
| **NDcPP22e:FAU_GEN.1** | | |
| **NDcPP22e:FAU_GEN.2** | | |
| **NDcPP22e:FAU_STG.1** | | |
| **NDcPP22e:FAU_STG_EXT.1** | | |
| **NDcPP22e:FCS_CKM.1** | | |
| **NDcPP22e:FCS_CKM.2** | | |
| **NDcPP22e:FCS_CKM.4** | | |
| **NDcPP22e:FCS_COP.1/DataEncryption** | | |
| **NDcPP22e:FCS_COP.1/Hash** | | |
| **NDcPP22e:FCS_COP.1/KeyedHash** | | |
| **NDcPP22e:FCS_COP.1/SigGen** | | |
| **NDcPP22e:FCS_RBG_EXT.1** | | |
| **NDcPP22e:FCS_SSHS_EXT.1** | Failure to establish an SSH session. | Reason for failure. |
| **NDcPP22e:FCS_TLSC_EXT.1** | Failure to establish a TLS Session. | Reason for failure. |
| **NDcPP22e:FIA_AFL.1** | Unsuccessful login attempt limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_PMG_EXT.1** | | |
| **NDcPP22e:FIA_UAU.7** | | |
| **NDcPP22e:FIA_UAU_EXT.2** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_UIA_EXT.1** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_X509_EXT.1/Rev** | Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store | Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| **NDcPP22e:FIA_X509_EXT.2** | | |
| **NDcPP22e:FMT_MOF.1/ManualUpdate** | Any attempt to initiate a manual update. | |
| **NDcPP22e:FMT_MTD.1/CoreData** | | |
| **NDcPP22e:FMT_MTD.1/CryptoKeys** | | |
| **NDcPP22e:FMT_SMF.1** | All management activities of TSF data. | |
| **NDcPP22e:FMT_SMR.2** | | |
| **NDcPP22e:FPT_APW_EXT.1** | | |

| | | |
|---|---|---|
| **NDcPP22e:FPT_SKP_EXT.1** | | |
| **NDcPP22e:FPT_STM_EXT.1** | Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| **NDcPP22e:FPT_TST_EXT.1** | | |
| **NDcPP22e:FPT_TUD_EXT.1** | Initiation of update; result of the update attempt (success or failure). | |
| **NDcPP22e:FPT_TUD_EXT.2** | Failure of update. | Reason for failure (including identifier of invalid certificate). |
| **NDcPP22e:FTA_SSL.3** | The termination of a remote session by the session locking mechanism. | |
| **NDcPP22e:FTA_SSL.4** | The termination of an interactive session. | |
| **NDcPP22e:FTA_SSL_EXT.1** | (if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism. | |
| **NDcPP22e:FTA_TAB.1** | | |
| **NDcPP22e:FTP_ITC.1** | Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| **NDcPP22e:FTP_TRP.1/Admin** | Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions. | |

**Table 3Audit Events**

#### 5.1.1.2 Audit Data Generation (MACsec) (MACSEC10:FAU_GEN.1/MACSEC)

**MACSEC10:FAU_GEN.1.1/MACSEC**

The TSF shall be able to generate an audit record of the following auditable events:
a. Start-up and shutdown of the audit functions;
b. All auditable events for the not specified level of audit;
c. All administrative actions;
d. Specifically defined auditable events listed in the Auditable Events table (Table 4)

**MACSEC10:FAU_GEN.1.2/MACSEC**

The TSF shall record within each audit record at least the following information:
a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-Module/ST, information specified in column three of the Auditable Events table (**Table 4**).

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_MACSEC_EXT.1 | Session establishment | Secure Channel Identifier (SCI) |
| FCS_MACSEC_EXT.3 | Creation and update of SAK | Creation and update times |
| FCS_MACSEC_EXT.4 | Creation of CA | Connectivity Association Key Names (CKNs) |
| FPT_RPL.1 | Detected replay attempt | None |

**Table 4 MACsec Audit Events**

#### 5.1.1.3 User identity association (NDcPP22e:FAU_GEN.2)

**NDcPP22e:FAU_GEN.2.1**

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

#### 5.1.1.4 Protected audit trail storage (NDcPP22e:FAU_STG.1)

**NDcPP22e:FAU_STG.1.1**

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

**NDcPP22e:FAU_STG.1.2**

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

#### 5.1.1.5 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1)

**NDcPP22e:FAU_STG_EXT.1.1**

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

**NDcPP22e:FAU_STG_EXT.1.2**

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

**NDcPP22e:FAU_STG_EXT.1.3**

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

### 5.1.2 Cryptographic support (FCS)

#### 5.1.2.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

**NDcPP22e:FCS_CKM.1.1**

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

### 5.1.2.2 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

**NDcPP22e:FCS_CKM.2.1**

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

### 5.1.2.3 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

**NDcPP22e:FCS_CKM.4.1**

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

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

that meets the following: No Standard.

### 5.1.2.4 Cryptographic Operation (AES-CMAC Keyed Hash Algorithm) (MACSEC10:FCS_COP.1/CMAC)

**MACSEC10:FCS_COP.1.1/CMAC**

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm AES-CMAC and cryptographic key sizes [***128, 256***] bits and message digest size of 128 bits that meets the following: NIST SP 800-38B.

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

**NDcPP22e:FCS_COP.1.1/DataEncryption**

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

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

**NDcPP22e:FCS_COP.1.1/Hash**

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

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

**NDcPP22e:FCS_COP.1.1/KeyedHash**

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-***

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

#### 5.1.2.8 Cryptographic Operation (MACsec AES Data Encryption and Decryption) (MACSEC10:FCS_COP.1/MACSEC)

**MACSEC10:FCS_COP.1.1/MACSEC**

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES used in AES Key Wrap, GCM and cryptographic key sizes [***128, 256***] bits that meets the following: AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772.

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

**NDcPP22e:FCS_COP.1.1/SigGen**

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [
***- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits and 3072 bits],***
***- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]***]
that meet the following:
[***- For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,***
***- For ECDSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4***].

#### 5.1.2.10 MACsec (MACSEC10:FCS_MACSEC_EXT.1)

**MACSEC10:FCS_MACSEC_EXT.1.1**

The TSF shall implement MACsec in accordance with IEEE Standard 802.1AE-2018.

**MACSEC10:FCS_MACSEC_EXT.1.2**

The TSF shall derive a Secure Channel Identifier (SCI) from a peer's MAC address and port to uniquely identify the originator of an MPDU.

**MACSEC10:FCS_MACSEC_EXT.1.3**

The TSF shall reject any MPDUs during a given session that contain an SCI other than the one used to establish that session.

**MACSEC10:FCS_MACSEC_EXT.1.4**

The TSF shall permit only EAPOL (Port Access Entity (PAE) EtherType 88-8E), MACsec frames (EtherType 88-E5), and [***no other frame types***] and shall discard others.

#### 5.1.2.11 MACsec Integrity and Confidentiality (MACSEC10:FCS_MACSEC_EXT.2)

**MACSEC10:FCS_MACSEC_EXT.2.1**

The TOE shall implement MACsec with support for integrity protection with a confidentiality offset of [***0, 30, 50***].

**MACSEC10:FCS_MACSEC_EXT.2.2**

The TSF shall provide assurance of the integrity of protocol data units (MPDUs) using an Integrity Check Value (ICV) derived with the SAK.

**MACSEC10:FCS_MACSEC_EXT.2.3**

The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a Connectivity Association Key (CAK) using a KDF.

### 5.1.2.12 MACsec Randomness (MACSEC10:FCS_MACSEC_EXT.3)

**MACSEC10:FCS_MACSEC_EXT.3.1**

The TSF shall generate unique Secure Association Keys (SAKs) using [***key derivation from Connectivity Association Key (CAK) per section 9.8.1 of IEEE 802.1X-2010***] such that the likelihood of a repeating SAK is no less than 1 in 2 to the power of the size of the generated key.

**MACSEC10:FCS_MACSEC_EXT.3.2**

The TSF shall generate unique nonces for the derivation of SAKs using the TOE's random bit generator as specified by FCS_RBG_EXT.1.

### 5.1.2.13 MACsec Key Usage (MACSEC10:FCS_MACSEC_EXT.4)

**MACSEC10:FCS_MACSEC_EXT.4.1**

The TSF shall support peer authentication using pre-shared keys (PSKs) [***no other method]***.

**MACSEC10:FCS_MACSEC_EXT.4.2**

The TSF shall distribute SAKs between MACsec peers using AES key wrap as specified in FCS_COP.1/MACSEC.

**MACSEC10:FCS_MACSEC_EXT.4.3**

The TSF shall support specifying a lifetime for CAKs.

**MACSEC10:FCS_MACSEC_EXT.4.4**

The TSF shall associate Connectivity Association Key Names (CKNs) with SAKs that are defined by the KDF using the CAK as input data (per IEEE 802.1X-2010, Section 9.8.1).

**MACSEC10:FCS_MACSEC_EXT.4.5**

The TSF shall associate CKNs with CAKs. The length of the CKN shall be an integer number of octets, between 1 and 32 (inclusive).

### 5.1.2.14 MACsec Key Agreement (MACSEC10:FCS_MKA_EXT.1)

**MACSEC10:FCS_MKA_EXT.1.1**

The TSF shall implement Key Agreement Protocol (MKA) in accordance with IEEE 802.1X-2010 and 802.1Xbx-2014.

**MACSEC10:FCS_MKA_EXT.1.2**

The TSF shall provide assurance of the integrity of MKA protocol data units (MKPDUs) using an Integrity Check Value (ICV) derived from an Integrity Check Value Key (ICK).

**MACSEC10:FCS_MKA_EXT.1.3**

The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a CAK using a KDF.

**MACSEC10:FCS_MKA_EXT.1.4**

The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and [***MKA Hello Time limit of 2 seconds***]. (TD0882 applied)

**MACSEC10:FCS_MKA_EXT.1.5**

The Key Server shall refresh a SAK when it expires. The Key Server shall distribute a SAK by [***pairwise CAKs that are PSKs***].

**MACSEC10:FCS_MKA_EXT.1.6**

The Key Server shall distribute a fresh SAK whenever a member is added to or removed from the live membership of the CA.

**MACSEC10:FCS_MKA_EXT.1.7**

The TSF shall validate MKPDUs according to IEEE 802.1X-2010 Section 11.11.2. In particular, the TSF shall discard without further processing any MKPDUs to which any of the following conditions apply:

a. The destination address of the MKPDU was an individual address

b. The MKPDU is less than 32 octets long

c. The MKPDU comprises fewer octets than indicated by the Basic Parameter Set body length, as encoded in bits 4 through 1 of octet 3 and bits 8 through 1 of octet 4, plus 16 octets of ICV

d. The CAK Name is not recognized

If an MKPDU passes these tests, then the TSF will begin processing it as follows:

a. If the Algorithm Agility parameter identifies an algorithm that has been implemented by the receiver, the ICV shall be verified as specified in IEEE 802.1X-2010 Section 9.4.1.

b. If the Algorithm Agility parameter is unrecognized or not implemented by the receiver, its value can be recorded for diagnosis but the received MKPDU shall be discarded without further processing.

Each received MKPDU that is validated as specified in this clause and verified as specified in IEEE 802.1X-2010 Section 9.4.1 shall be decoded as specified in IEEE 802.1X-2010 Section 11.11.4.

#### 5.1.2.15 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

**NDcPP22e:FCS_RBG_EXT.1.1**

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

**NDcPP22e:FCS_RBG_EXT.1.2**

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

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

**NDcPP22e:FCS_SSHS_EXT.1.1**

The TSF shall implement the SSH protocol that complies with: RFC(s) 4251, 4252, 4253, 4254, [***4256, 4344, 5656, 6668, 8308 section 3.1***].

**NDcPP22e:FCS_SSHS_EXT.1.2**

The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [***password-based***]. (TD0631 applied)

**NDcPP22e:FCS_SSHS_EXT.1.3**

The TSF shall ensure that, as described in RFC 4253, packets greater than [***262,149***] bytes in an SSH transport connection are dropped.

**NDcPP22e:FCS_SSHS_EXT.1.4**

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

**NDcPP22e:FCS_SSHS_EXT.1.5**

The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521***] as its public key algorithm(s) and rejects all other public key algorithms.

**NDcPP22e:FCS_SSHS_EXT.1.6**

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

**NDcPP22e:FCS_SSHS_EXT.1.7**

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

**NDcPP22e:FCS_SSHS_EXT.1.8**

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

#### 5.1.2.17 TLS Client Protocol Without Mutual Authentication - per TD0670 & TD0790 (NDcPP22e:FCS_TLSC_EXT.1)

**NDcPP22e:FCS_TLSC_EXT.1.1**

The TSF shall implement [***TLS 1.2 (RFC 5246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:
[***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,***
***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***]
and no other ciphersuites.

**NDcPP22e:FCS_TLSC_EXT.1.2**

The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 section 6, IPv4 address in SAN***].

**NDcPP22e:FCS_TLSC_EXT.1.3**

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

**NDcPP22e:FCS_TLSC_EXT.1.4**

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

### 5.1.3 Identification and authentication (FIA)

#### 5.1.3.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

**NDcPP22e:FIA_AFL.1.1**

The TSF shall detect when an Administrator configurable positive integer within [**1-5**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

**NDcPP22e:FIA_AFL.1.2**

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

#### 5.1.3.2 Password Management - per TD0792 (NDcPP22e:FIA_PMG_EXT.1)

**NDcPP22e:FIA_PMG_EXT.1.1**

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', ['", '+', ",", "-", ".", "/", ":", ";", "<", "=", ">", "[", "\", "]", "_", "`", "{", "}", "<space>", and "~"***];

b) Minimum password length shall be configurable to between [**1**] and [**128**] characters.

### 5.1.3.3 Pre-Shared Key Composition (MACSEC10:FIA_PSK_EXT.1)

**MACSEC10:FIA_PSK_EXT.1.1**

The TSF shall use PSKs for MKA as defined by IEEE 802.1X-2010, [***no other protocols***].

**MACSEC10:FIA_PSK_EXT.1.2**

The TSF shall be able to [***accept***] bit-based PSKs.

### 5.1.3.4 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

**NDcPP22e:FIA_UAU.7.1**

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

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

**NDcPP22e:FIA_UAU_EXT.2.1**

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

### 5.1.3.6 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

**NDcPP22e:FIA_UIA_EXT.1.1**

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

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

**NDcPP22e:FIA_UIA_EXT.1.2**

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

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

**NDcPP22e:FIA_X509_EXT.1.1/Rev**

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:

o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

o Server certificates presented for 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.

o 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.

o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

**NDcPP22e:FIA_X509_EXT.1.2/Rev**

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

#### 5.1.3.8 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

**NDcPP22e:FIA_X509_EXT.2.1**

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

**NDcPP22e:FIA_X509_EXT.2.2**

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

### 5.1.4 Security management (FMT)

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

**NDcPP22e:FMT_MOF.1.1/ManualUpdate**

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

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

**NDcPP22e:FMT_MTD.1.1/CoreData**

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

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

**NDcPP22e:FMT_MTD.1.1/CryptoKeys**

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

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

**NDcPP22e:FMT_SMF.1.1**

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***hash comparison***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),***
***- Ability to modify the behavior of the transmission of audit data to an external IT entity,***
***- Ability to manage the cryptographic keys,***
***- Ability to configure the cryptographic functionality,***
***- Ability to configure thresholds for SSH rekeying,***
***- Ability to set the time which is used for time-stamps,***
***- Ability to configure the reference identifier for the peer,***
***- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors, - -***
***- Ability to import X509v3 certificates to the TOE's trust store,***
***- Ability to manage the trusted public keys database***].
(TD0631 applied)

#### 5.1.4.5 Specification of Management Functions (MACsec) - per TD0748 (MACSEC10:FMT_SMF.1/MACSEC)

**MACSEC10:FMT_SMF.1.1/MACSEC**

The TSF shall be capable of performing the following management functions related to MACsec functionality: Ability of a Security Administrator to:
- Manage a PSK-based CAK and install it in the device
- Manage the key server to create, delete, and activate MKA participants [***as specified in IEEE 802.1X-2020, Sections 9.13 and 9.16 (cf. MIB object ieee8021XKayMkaParticipant Entry) and section 12.2 (cf. function createMKA()***]
- Specify the lifetime of a CAK
- Enable, disable, or delete a PSK-based CAK using [***CLI management commands***]

[***Manage generation of a PSK-based CAK***].

#### 5.1.4.6 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2)

**NDcPP22e:FMT_SMR.2.1**

The TSF shall maintain the roles: - Security Administrator.

**NDcPP22e:FMT_SMR.2.2**

The TSF shall be able to associate users with roles.

**NDcPP22e:FMT_SMR.2.3**

The TSF shall ensure that the conditions
- The Security Administrator role shall be able to administer the TOE locally;
- The Security Administrator role shall be able to administer the TOE remotely

are satisfied.

### 5.1.5 Protection of the TSF (FPT)

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

**NDcPP22e:FPT_APW_EXT.1.1**

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

**NDcPP22e:FPT_APW_EXT.1.2**

The TSF shall prevent the reading of plaintext administrative passwords.

#### 5.1.5.2 Protection of CAK Data (MACSEC10:FPT_CAK_EXT.1)

**MACSEC10:FPT_CAK_EXT.1.1**

The TSF shall prevent reading of CAK values by administrators.

#### 5.1.5.3 Failure with Preservation of Secure State (MACSEC10:FPT_FLS.1)

**MACSEC10:FPT_FLS.1.1**

The TSF shall fail-secure when any of the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

#### 5.1.5.4 Replay Detection - per TD0746 (MACSEC10:FPT_RPL.1)

**MACSEC10:FPT_RPL.1.1**

The TSF shall detect replay for the following entities: MPDUs, MKA frames.

**MACSEC10:FPT_RPL.1.2**

The TSF shall perform discarding of the replayed data, logging of the detected replay attempt when replay is detected.

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

**NDcPP22e:FPT_SKP_EXT.1.1**

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

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

**NDcPP22e:FPT_STM_EXT.1.1**

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

**NDcPP22e:FPT_STM_EXT.1.2**

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

#### 5.1.5.7 TSF testing (NDcPP22e:FPT_TST_EXT.1)

**NDcPP22e:FPT_TST_EXT.1.1**

The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [

- **Check of flash access and content with CRC (integrity checks),**
- **Check of various FPGA devices access and sanity,**
- **Check of PCI bus and devices response,**
- **Sanity check of memory,**
- **Check of FANS,**
- **Crypto KAT/self-test**].

#### 5.1.5.8 Trusted update (NDcPP22e:FPT_TUD_EXT.1)

**NDcPP22e:FPT_TUD_EXT.1.1**

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [***the most recently installed version of the TOE firmware/software***].

**NDcPP22e:FPT_TUD_EXT.1.2**

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

**NDcPP22e:FPT_TUD_EXT.1.3**

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

### 5.1.6 TOE access (FTA)

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

**NDcPP22e:FTA_SSL.3.1**

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

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

**NDcPP22e:FTA_SSL.4.1**

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

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

**NDcPP22e:FTA_SSL_EXT.1.1**

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

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

**NDcPP22e:FTA_TAB.1.1**

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

### 5.1.7 Trusted path/channels (FTP)

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

**NDcPP22e:FTP_ITC.1.1**

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

**NDcPP22e:FTP_ITC.1.2**

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

**NDcPP22e:FTP_ITC.1.3**

The TSF shall initiate communication via the trusted channel for [**transmission of audit data**].

#### 5.1.7.2 Inter-TSF Trusted Channel (MACsec Communications) (MACSEC10:FTP_ITC.1/MACSEC)

**MACSEC10:FTP_ITC.1.1/MACSEC**

The TSF shall provide a communication channel between itself and a MACsec peer that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

**MACSEC10:FTP_ITC.1.2/MACSEC**

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

**MACSEC10:FTP_ITC.1.3/MACSEC**

The TSF shall initiate communication via the trusted channel for communications with MACsec peers that require the use of MACsec.

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

**NDcPP22e: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.

**NDcPP22e:FTP_TRP.1.2/Admin**

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

**NDcPP22e:FTP_TRP.1.3/Admin**

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

## 5.2 TOE Security Assurance Requirements

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

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

**Table 5 Assurance Components**

### 5.2.1 Development (ADV)

#### 5.2.1.1 Basic Functional Specification (ADV_FSP.1)

**ADV_FSP.1.1d**

The developer shall provide a functional specification.

**ADV_FSP.1.2d**

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

**ADV_FSP.1.1c**

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

**ADV_FSP.1.2c**

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

**ADV_FSP.1.3c**

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

**ADV_FSP.1.4c**

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

**ADV_FSP.1.1e**

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

**ADV_FSP.1.2e**

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

### 5.2.2 Guidance documents (AGD)

#### 5.2.2.1 Operational User Guidance (AGD_OPE.1)

**AGD_OPE.1.1d**

The developer shall provide operational user guidance.

**AGD_OPE.1.1c**

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

**AGD_OPE.1.2c**

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

**AGD_OPE.1.3c**

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

**AGD_OPE.1.4c**

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

**AGD_OPE.1.5c**

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

**AGD_OPE.1.6c**

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

**AGD_OPE.1.7c**

The operational user guidance shall be clear and reasonable.

**AGD_OPE.1.1e**

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

#### 5.2.2.2 Preparative Procedures (AGD_PRE.1)

**AGD_PRE.1.1d**

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

**AGD_PRE.1.1c**

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

**AGD_PRE.1.2c**

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

**AGD_PRE.1.1e**

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

**AGD_PRE.1.2e**

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

### 5.2.3 Life-cycle support (ALC)

#### 5.2.3.1 Labelling of the TOE (ALC_CMC.1)

**ALC_CMC.1.1d**

The developer shall provide the TOE and a reference for the TOE.

**ALC_CMC.1.1c**

The TOE shall be labelled with its unique reference.

**ALC_CMC.1.1e**

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

#### 5.2.3.2 TOE CM Coverage (ALC_CMS.1)

**ALC_CMS.1.1d**

The developer shall provide a configuration list for the TOE.

**ALC_CMS.1.1c**

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

**ALC_CMS.1.2c**

The configuration list shall uniquely identify the configuration items.

**ALC_CMS.1.1e**

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

### 5.2.4 Tests (ATE)

#### 5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

**ATE_IND.1.1d**

The developer shall provide the TOE for testing.

**ATE_IND.1.1c**

The TOE shall be suitable for testing.

**ATE_IND.1.1e**

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

**ATE_IND.1.2e**

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

### 5.2.5 Vulnerability assessment (AVA)

#### 5.2.5.1 Vulnerability Survey (AVA_VAN.1)

**AVA_VAN.1.1d**

The developer shall provide the TOE for testing.

**AVA_VAN.1.1c**

The TOE shall be suitable for testing.

**AVA_VAN.1.1e**

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

**AVA_VAN.1.2e**

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

**AVA_VAN.1.3e**

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

# 6. TOE Summary Specification

This chapter describes the security functions:

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

## 6.1 Security audit

**NDcPP22e:FAU_GEN.1| MACSEC10:FAU_GEN.1/MACSEC| NDcPP22e:FAU_GEN.2**

The TOE generates an audit record whenever an auditable event occurs. Auditable events include the start-up and shut-down of the audit function, and all administrative actions. The administrative actions include login and logout by administrators, all changes to TOE configuration, all actions (generating, importing, exporting, renaming, moving, and deleting) of cryptographic keys, and all changes in passwords. The TOE also generates an audit record for each event stated in Table 3 and **Table 4**.

Each of the events specified in the audit record is in enough detail to identify the user for which the event is associated (e.g., user identity, IP address), when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred. Cryptographic keys are identified by a unique name.

The date and time information for any audit event is recorded by the TOE as part of each audit record to ensure the timing of the event can be unambiguously determined from the data contained in the audit record. The representation of date and time information recorded for each event allows unanimous determination of at day, month and year information for the date and hours, minutes and second information for the time.

**NDcPP22e:FAU_STG.1**

Only authorized administrators may view audit records using the CLI which is the sole interface to the management functions of the TOE. No capability to modify the audit records is implemented in the CLI.

**NDcPP22e:FAU_STG_EXT.1**

Audit records are stored persistently on the local file system to which only the administrator has access. The TOE is a standalone component that stores audit data locally. The TOE leverages the log rotate services to create a circular local log buffer where by default newer messages overwrite older messages after the buffer is full. The log rotate service, archives and rotates the current log file when it reaches a size of 2 Megabytes. The TOE will keep up to 10 archived log files, so the oldest archived file will be deleted as the newest log file is filled and archived.

The Security Administrator can configure the TOE to transfer the audit data to an external audit server. The audit logs are sent to the external syslog server via TLS in real-time.

## 6.2 Cryptographic support

The TOE supports a range of cryptographic services using the as the Ciena Cryptographic Library SAOS 10.x version1.0. The following functions have been CAVP tested:

| Functions | Standards | Certificates |
|---|---|---|
| **Asymmetric key generation and key verification (FCS_CKM.1)** | | |

| Functions | Standards | Certificates |
|---|---|---|
| RSA Schemes (2048 bit, 3072 bit) | FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 | A2492 |
| ECC Schemes (ECDSA P-256, P-384, P-521 curves) | FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4” | A2492 |
| FFC Schemes (‘safe-prime’ groups) | NIST SP 800-56A Revision 3 ,_“Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526 | Tested with known good implementation |
| **Key Establishment (FCS_CKM.2)** | | |
| RSA- based scheme | RSAES-PKCS1-v1_5, Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1” | Tested with known good implementation |
| Elliptic curve-based scheme | NIST SP 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” | A2492 |
| Finite field-based scheme | NIST Special Publication 800-56A, Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and<br>RFC 3526 | Tested with known good implementation |
| **Encryption/Decryption (FCS_COP.1/DataEncryption)** | | |
| AES in CBC mode (128, 256 bits) | AES as specified in ISO 18033-3.<br>CBC as specified in ISO 10116. | A2492 |
| AES in CTR mode (128, 256 bits) | AES as specified in ISO 18033-3.<br>CTR as specified in ISO 10116. | A2492 |
| AES in GCM mode (128, 256 bits) | AES as specified in ISO 18033-3.<br>GCM as specified in ISO 19772. | A2492 |
| **MACsec AES Data Encryption and Decryption (FCS_COP.1/MACSEC)** | | |
| AES-KW (128, 256 bits) | AES as specified in ISO 18033-3.<br>AES Key Wrap as specified in NIST SP 800-38F. | A2492 |

| Functions | Standards | Certificates |
|---|---|---|
| AES in GCM mode (128, 256 bits) | AES as specified in ISO 18033-3.<br>GCM as specified in ISO 19772. | AES 4550 |
| **Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)** | | |
| RSA Digital Signature Algorithm (rDSA) (2048 AND 3072-bit modulus) | 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, | A2492 |
| ECDSA schemes (P-256, P-384, P-521) | FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4 | A2492 |
| **Cryptographic hashing (FCS_COP.1/Hash)** | | |
| SHA-1 (digest size 160 bits)<br>SHA-256 (digest size 256 bits)<br>SHA-384 (digest size 384 bits)<br>SHA-512 (digest size 512 bits) | ISO/IEC 10118-3:2004 | A2492 |
| **Keyed-hash message authentication (FCS_COP.1/KeyedHash)** | | |
| HMAC-SHA-1 (key size 160, digest size 160);<br>HMAC-SHA-256 (key size 256 bits, digest size 256 bits);<br>HMAC-SHA-384 (key size 384 bits, digest size 384 bits);<br>HMAC-SHA-512 (key size 512 bits, digest size 512 bits) | ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2" | A2492 |
| **AES-CMAC Keyed Hash Algorithm (FCS_COP.1/CMAC)** | | |
| AES-CMAC (128, 256 bits) | AES as specified in ISO 18033-3.<br>CMAC as specified in NIST SP 800-38B. | A2492 |
| **Random bit generation (FCS_RBG_EXT.1/ARMA53, FCS_RBG_EXT.1/ARMA72, FCS_RBG_EXT.1/Intel)** | | |

| Functions | Standards | Certificates |
|---|---|---|
| CTR-DRBG (AES-256) – 256 bits entropy; | ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions" | A2492 |

**NDcPP22e:FCS_CKM.1**

The TOE implements generation of asymmetric cryptographic keys using the following asymmetric schemes:

- RSA schemes using cryptographic key sizes of 2048 bits and 3072 bits as defined in FIPS PUB 186-4,
- ECC schemes using NIST curves P-256, P-384 and P-521 as defined in FIPS PUB 186-4, and
- Diffie-Hellman Groups 14 (2048-bit MODP) and 16 (4096-bit MODP)

The asymmetric cryptographic keys generated by the above methods are used by the protocols the TOE implements for trusted path and trusted channels as follows:

| | TLS | SSH |
|---|---|---|
| RSA | 2048 bits, 3072 bits | 2048 bits, 3072 bits |
| ECC | secp256r1, secp384r1, secp521r1 | nistp256, nistp384, nistp521 |
| FFC (DH Groups) | N/A | Group 14, Group 16 |

**NDcPP22e:FCS_CKM.2**

The TOE implements RSA, ECC and FFC (DH Groups) for the key establishment.

RSA and ECC are used in TLS and SSH. TLS is used for protecting the communication between the TOE and the audit server. SSH is used for protecting the remote management session between the remote management workstation and the TOE.

ECC is additionally used for the verification of the digital signatures in public key certificates of the TLS peer entities.

DH Groups 14 and 16 are used for implementing SSH which protects the remote management session between the remote management workstation and the TOE.

**NDcPP22e:FCS_CKM.4**

The TOE destroys plaintext cryptographic keys stored in the volatile storage by a single overwrite with zeroes. Plaintext keys stored in the non-volatile storage are destroyed by the SAOS overwriting the storage location of the key with a single overwrite of zeroes. The destruction of each cryptographic key and Critical Security Parameter (CSP) is summarized in the following table.

| Keys/CSPs | Purpose | Storage Location | Method of Zeroization |
|---|---|---|---|
| SSH Server Host Keys | The SSH server host keys to identify ssh server | Non-volatile storage/file system | Overwrite with zeros to clear cache and read verify, then erase file |
| SSH session keys | Keys exchanged for protecting the confidentiality of the remote administration session | Volatile storage | Openssh package is used but all keys are overwritten with zeros before freeing memory |

| | | | |
|---|---|---|---|
| SSH PKA | Public key authentication for remote administration over SSH | Non-volatile storage | Overwrite with zeros to clear cache and read verify, then erase file |
| X509 certificate with keys | For TLS connections | Non-volatile storage/file system | Overwrite with zeros to clear cache and read verify, then erase file |
| Local user password | User login | Non-volatile storage/file system (shadow file) | Erase file. Password is hashed with sha512 and the password file is only readable by root |
| TLS session HMAC keys | For TLS connections with Audit server | Volatile storage | OpenSSL package is used as infrastructure but all keys are overwritten with zeros before freeing memory |
| MACsec SAK | For securing MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |
| MACsec CAK | For deriving the SAK in MACsec Connections | Non-volatile storage/file system | Overwritten with zeros when deleted by administrator command. |
| MACsec Key Encryption Key (KEK) | For securing transport of the SAK in MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |
| MACsec Integrity Check Key (ICK) | For verifying the integrity of data in MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |

**Table 6 Keys and CSPs**

**MACSEC10:FCS_COP.1/CMAC**

The TOE supports keyed-hash message authentication in accordance with AES-CMAC algorithm with key sizes 128 bits and 256 bits and the message digest size supported is 128 bits. The algorithm conforms to NIST SP 800-38B.

**NDcPP22e:FCS_COP.1/DataEncryption**

The TOE implements symmetric encryption and decryption using AES in CBC, GCM and CTR modes. Key sizes of 128 and 256 bits are implemented. AES encryption and decryption is used by TLS, SSH, and MACsec protocols.

**NDcPP22e:FCS_COP.1/Hash**

The TOE implements cryptographic message digest (hash value) computation using SHA-1, SHA-256, SHA-384, and SHA-512 with message digest sizes of 160 bits, 256 bits, 384 bits, and 512 bits respectively. The hashing algorithms are used in SSH and TLS connections for secure communications.

The TOE uses message digests for the following functions:

| Function | SHA-1 | SHA-256 | SHA-384 | SHA-512 |
|---|---|---|---|---|
| Digital signature computation |  | X | X | X |
| Digital Signature verification | X | X | X | X |
| TLS HMAC | X | X | X |  |
| SSH HMAC | X | X |  | X |
| SSH PKA |  | X |  |  |
| Password storage |  |  |  | X |

**NDcPP22e:FCS_COP.1/KeyedHash**

For its HMAC implementations, the TOE accepts all key sizes of 160, 256, 384, & 512; supports SHA sizes SHA-1, 256, 384, & 512, utilizes the specified block size (512 for SHA-1 and 256, and 1024 for SHA-384 & 512); and outputs MAC lengths of 160, 256, 384, and 512.

**MACSEC10:FCS_COP.1/MACSEC**

The TOE performs AES key wrap as specified in AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772.

**NDcPP22e:FCS_COP.1/SigGen**

The TOE generates and verifies digital signatures with RSA Digital Signature Algorithm and cryptographic key sizes (modulus) 2048 bits and 3072 bits.

The TOE also generates and verifies digital signatures with ECC using key sizes 256 bits, 384 bits, and 512 bits for NIST curves P-256, P-384, and P-521.

**MACSEC10:FCS_MACSEC_EXT.1**

The TOE implements MACsec in accordance with IEEE 802.1AE-2018. The TOE derives a Secure Channel Identifier (SCI) from a peer's MAC address and port to uniquely identify the originator of a MACsec Protocol Data Unit (MPDU) and rejects any MPDUs that do not contain the identifier.  Only EAPOL (PAE EtherType 88-8E) and MACsec frames (EtherType 88-E5) are permitted and others are rejected.

**MACSEC10:FCS_MACSEC_EXT.2**

The TOE implements the MACsec requirement for integrity protection with the confidentiality offset of 0, 30, or 50. The TOE derives the ICV from a CAK using KDF, using the SCI as the most significant bits of the IV and the 32 least significant bits of the PN as the IV. The supported ICV length is 16 octets. An ICV derived with the SAK is used to provide assurance of the integrity of MPDUs. The ICV is generated by a pre-shared key configured via the CLI.

**MACSEC10:FCS_MACSEC_EXT.3**

A SAK and CAK are derived by a pre-shared key configured via the CLI. The pre-shared key must be 128 bits when using gcm-aes-128 and 256 bits when using gcm-aes-256. The TOE’s random bit generator is used for creating these unique nonces.

**MACSEC10:FCS_MACSEC_EXT.4**

The TOE ensures MACsec peer authentication using pre-shared keys. The TOE uses AES Key Wrap to distribute the SAKs between peers using aes-128-cmac or aes-256-cmac. The TOE associates Connectivity Association Key Names (CKNs) with CAKs. The length of the CKN shall be an integer number of octets, between 1 and 32 (inclusive). The TOE does not support group CAKs.

**MACSEC10:FCS_MKA_EXT.1**

The TOE implements Key Agreement Protocol (MKA) in accordance with IEEE 802.1X-2010. The TOE supports the data delay protection to provide security against delay attack. The TOE enforces an MKA Lifetime Timeout limit of 6.0 seconds and a MKA Hello Time limit of 2 seconds. Data delay protection is provided by discarding any Data frame which is received out-of-order. Data received in only in Strict-order is accepted by hardware; all others are discarded by the hardware. The TOE verifies the integrity of MKA protocol data units using an ICV derived from the ICK. The ICK is derived from the CAK using KDF (AES-CMAC). The ICV is checked on the reception of each MKA PDU.

**NDcPP22e:FCS_RBG_EXT.1**

The TOE implements a CTR_DRBG (AES-256) implemented by OpenSSL and seeded by 256 bits of data read from the Kernel DRBG. The entropy sources is from the Linux kernel v5.4 functions which accumulate and make available to other processes entropy from CPU jitter and CPU interrupts.

**NDcPP22e:FCS_SSHS_EXT.1**

The TOE implements a SSH Server for remote administrators to connect securely to the TOE and use the CLI from a remote management station. The TOE implementation of SSHv2 is in compliance with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668, 8308 section 3.1, 8332.

The TOE implements both public key authentication and password-based authentication. Public key authentication methods supported are ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521. When a user presents a public key for authentication, TOE checks the presented public key matches one that is stored within the server's authorized_keys file. Any other authentication algorithm requests are rejected.

The TOE examines all packets for size and drops any packets greater than 262,149 bytes and drop in accordance with RFC 4253.

For symmetric encryption, the TOE allows aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, and aes256-gcm@openssh.com. Requests for any other algorithms are rejected.

For message authentication, the TOE allows hmac-sha1, hmac-sha2-256, hmac-sha2-512 and implicit. Requests for any other algorithms is rejected. Message authentication algorithm implicit is used for the @openssh.com symmetric encryption algorithms.

The SSHv2 implementation of the TOE enforces to only allow the diffie-hellman-group14-sha1, ecdh-sha2-nistp256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521 key exchange methods.

The TOE is capable of rekeying the SSH connection. The rekeying occurs if a session is longer than one hour or more than one gigabytes of data has been transmitted with one key. The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey.

**NDcPP22e:FCS_TLSC_EXT.1**

The TOE implements a TLS Client which supports TLS 1.2 (RFC 5246) and rejects all other TLS and SSL versions. The following ciphersuites are implemented:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The TOE uses TLS for all trusted channels. Peer entities are authenticated with the X.509 certificates. Mutual authentication is not supported. The trusted channel is established when the peer certificate is valid. The TOE verifies that the presented identifier matches the reference identifier in order to establish the connection.

The TOE supports the use of FQDN and IPv4 addresses as reference identifiers within the certificate's Common Name (CN) or Subject Alternative Name (SAN) extension. The users can configure the IP address or FQDN and the TOE will verify certificate fields against locally configured peer DNS name or IP address (Subject Name Authorization) as per RFC6125 Section 6.

The TOE supports SAN extension and checks SAN extension over CN when present. The TOE ignores CN when SAN is present. When SAN is not present, the TOE falls back to CN check. FQDN is supported in both SAN and CN while IP address is only supported in SAN. Wildcards are supported for DN names. The TLS client does not support certificate pinning.

The TLS client will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the NIST curves secp256r1, secp384r1 and secp521r1. The non-TOE server can choose to negotiate the elliptic curve from this set for any of the mutually negotiable elliptic curve cipher suites. The TOE will validate the server's certificate according to FIA_X509_EXT.1/Rev. If the server certificate is invalid, the connection will not be established.

## 6.3 Identification and authentication

**NDcPP22e:FIA_AFL.1**

The TOE maintains a counter of consecutive failed authentication attempts for each user. The counter tracks the number of failed authentication attempts for all remote authentication attempts. Local authentication attempts from the console are not tracked by the counter.

When an authentication fails, the counter value is incremented. When an authentication succeeds, the counter value is reset. The maximum number of allowed consecutive authentication attempts may be set by the administrator of the TOE. When the maximum number is reached, the TOE shall lock the account and start a session lockout timer for the account. Once the lockout timer expires, the TOE shall unlock the account. The duration of the lockout may be set by the administrator of the TOE.

Accounting locking only applies to remote authentication attempts. Even if an account is locked, the same account may still be used from console if the user is successfully authenticated. This ensures that at no time shall the TOE be in a state where each administrator is locked out and no administrator access to the TOE is possible.

**NDcPP22e:FIA_PMG_EXT.1**

Each user of the TOE may choose his/her password. To ensure high quality passwords, the TOE implements quality criteria which each password must meet. The criteria are implemented using the alphabet used for the password selection and the minimum length of the passwords.

The passwords must be expressed using the standard Linux alphabet allowed for passwords. The alphabet includes upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "'", "+", ",", "-", ".", "/", ":", ";", "<", "=", ">", "[", "\", "]", "_", "`", "{", "}", "<space>", and "~".

Each password must also be of the minimum length allowed by the TOE. The minimum length of a password may be configured by the administrator and can be any integer value between 1 and 128 (inclusive).

**MACSEC10:FIA_PSK_EXT.1**

The TOE supports the use of pre-shared keys for MKA as defined by IEEE 802.1X. The pre-shared keys are not generated by the TOE but rather the TOE will accept bit based pre-shared keys.

**NDcPP22e:FIA_UAU.7**

Local authentication is echoless, i.e., the TOE does not display on the console any characters when a password is entered.

**NDcPP22e:FIA_UAU_EXT.2**

The TOE uses local password-based authentication.

**NDcPP22e:FIA_UIA_EXT.1**

The TOE requires all users to be successfully identified and authenticated prior to assigning them to the role administrator and granting them access to the TOE. The TOE displays an access banner to each user prior at the identification and authentication window. Successful identification and authentication are required for each subsequent administrative access to the TOE.

Administrators may access the TOE locally from a console connected to the serial port of a USB-C port of the TOE, or remotely over a SSH connection.

For local access, the TOE prompts the user to enter a username and password. The TOE compares the entered password to the reference password stored for the user. If the verification succeeds and the user is allowed to enter the role administrator, the TOE assigns the user to the role administrator and grants access to the CLI. If the username does not exist or the password is incorrect, the TOE denies the access and returns to the authentication window to request a username and password.

For remote access, the TOE may be configured to require RSA public key authentication or password-based authentication. The remote access is implemented using SSH. Successful authentication occurs when either the cryptographic authentication protocol is successfully completed between the TOE and the remote management workstation, or the password verification succeeds in a manner identical to the authentication for local access.

If the authentication is successful and the user is granted access to the role administrator, the TOE establishes a SSH connection between the remote management workstation and the TOE and grants the user administrator rights to the TOE (i.e., makes available the CLI). If the authentication fails, the TOE denies access and returns to the authentication windows.

**NDcPP22e:FIA_X509_EXT.1/Rev**

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication of external TLS peers.

The TOE validates certificates in accordance with the following rules.  The TOE performs the same revocation checking on certificates regardless of whether it receives a full certificate chain or only a leaf certificate.

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TOE validates 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 TOE validates the revocation status of the certificate using Online Certificate Status Protocol (OCSP) as specified in RFC 6960.
- The TOE validates the extendedKeyUsage field according to the following rules:
  - Server certificates presented for TLS must 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 must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.

- The TOE will only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Certificate validity is checked on each certificate authentication. If the validation of the certificate fails because the OCSP Server cannot be connected to, the certificate shall not be accepted.

**NDcPP22e:FIA_X509_EXT.2**

The TOE uses X.509v3 certificates as defined by RFC 5280 for the authentication of TLS peer entities. Certificates are used to authenticate and establish secure communication channel for Syslog servers. The TOE supports RSA based certificates and ECC based certificate in PKCS#12.

The TOE allows each TLS service to be configured with its own certificate in a TLS profile. Once a certificate is configured for TLS Syslog, that certificate will be used for all TLS Syslog collector server connection authentication.

The TOE will check the validity of the TLS Server certificate prior to establishing a TLS connection with the TLS server. The certificate validation is determined based on reference ID verification, certificate path, extendedKeyUsage field, certificate expiry date and the certificate revocation status.

If the TOE is unable to establish a connection to OCSP responder to determine the validity of a certificate, the TOE will not accept the certificate and will not establish the connection.

If a certificate is deemed invalid, the TOE will not accept the certificate and therefore, not establish the connection.

## 6.4 Security management

**NDcPP22e:FMT_MOF.1/ManualUpdate**

The Vendor does occasionally publish software updates for the TOE to address security flaws or other bugs in the software. The TOE implements a function which allows administrators to install the software upgrades on the TOE.

The CLI of the TOE also implements a command show software which allows the administrator to query the currently executing version of the TOE software to determine whether it requires upgrading.

The Security Administrator may obtain the software image from the Ciena website and place it on a trusted file server. The TOE may then connect to the file server for the installation of the software upgrade. The installation of the upgrade is by the CLI command software install:

```
software install hash-algorithm sha256 hash-value url 'url information'
```

where hash-value is the digest value for the software package, published by Ciena on the software description file at the Ciena web site.

**NDcPP22e:FMT_MTD.1/CoreData**

The TOE requires successful identification and authentication of each administrator prior to granting them access to the TOE. Access to the TOE is granted by making available to the user a shell in which the user can execute CLI commands. Without access to the shell, the CLI is not accessible to the user and, consequently, administrator accesses are not possible. There are no management functions other than those accessible through the CLI.

The only access the TOE allows prior to the successful identification and authentication of the user is the access banner displayed at each login prompt.

**NDcPP22e:FMT_MTD.1/CryptoKeys**

The Security Administrator has the ability to configure the authentication keys TLS functionality and can modify, generate, and delete the key for SSH.

The TOE restricts the ability to manage SSH (session keys), TLS (session keys), any configured X.509 CA certificates and configuring a CAK for MACsec to the security administrators.

**NDcPP22e:FMT_SMF.1**

The TOE implements a management interface for the Security Administrators to configure the TOE. The management interface is a Command Line Interface (CLI) which may be accessed locally from a management workstation connected to the TOE on the console or USB-C interface, or from a remote management workstation connected to the TOE network management port over SSH.

The Security Administrator may use the CLI to manage the TOE locally and remotely. The CLI implements the following management functions:

- 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 authenticity of the updates using prior to installation,
- Ability to configure the authentication failure handling,
- Ability to configure the audit behavior of the TOE (including the  ability to modify the transmission of audit data to an external audit server),
- Ability to manage cryptographic keys,
- Ability to configure the cryptographic functionality,
- Ability to configure thresholds for SSH rekeying,
- Ability to set the time locally,
- Ability to configure the reference identifier for the peer
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,
- Ability to import X.509v3 certificates to the TOE's trust store, and
- Ability to manage the trusted public keys database.

**MACSEC10:FMT_SMF.1/MACSEC**

the TOE provides the Security Administrator the ability to:

- Generate a PSK-based CAK and install it in the device.
- Manage the Key Server to create, delete, and activate MKA participants as specified in 802.1X, sections 9.13 and 9.16 (cf. MIB object ieee8021XKayMkaParticipantEntry) and section.12.2 (cf. function createMKA)
- Specify a lifetime of a CAK
- Enable, disable, or delete a PSK-based CAK using cli commands

**NDcPP22e:FMT_SMR.2**

The TOE only implements a single role: Security Administrator. By default, the system has the following three pre-defined NACM groups:

- Limited (read only)
- Admin (can make significant system changes and modify the configuration, but cannot modify user accounts or authorizations)
- Super (can make significant system changes and modify the configuration, including user accounts or authorizations)

The evaluated configuration supports only one administrative role, Security Administrator. Users that belong to “Super” or “admin” groups have administrative privileges and assume the role of Security Administrator. The TOE also supports a single non-administrative role: Read-Only User. Users that belong to “Limited” group have read-only privileges. Read-Only User cannot make any changes to the TOE configuration.

## 6.5 Protection of the TSF

**NDcPP22e:FPT_APW_EXT.1**

All passwords are stored by the TOE hashed and salted using SHA-512. The storage and management of the passwords is implemented using the standard Linux Pluggable Authentication Mechanism (PAM) functions. There is no interface for administrators to read encrypted passwords.

**MACSEC10:FPT_CAK_EXT.1**

As part of configuration, the administrator enters the CAK but there is no interface to query it back. When the administrator issues a *show configuration* command, the CAK is not disclosed.

**MACSEC10:FPT_FLS.1**

If the TOE encounters a self-test failure, failure of integrity check of the TSF executable image, or failure of noise source health tests it will shutdown. The TOE will not boot as long as it has a failure.

**MACSEC10:FPT_RPL.1**

The TOE detects and logs all attempts to replay MPDUs and MKA frames by verifying the packet number (PN). If the received PN is lower than the current PN, this indicates to the TOE a replay attempt, and the packet is discarded.

**NDcPP22e:FPT_SKP_EXT.1**

During the setup and configuration of the TOE when cryptographic keys are generated, the TOE stores all private keys in a secure directory that is not readily accessible to administrators. Storage and destruction of each CSP and cryptographic key is summarized in

| Keys/CSPs | Purpose | Storage Location | Method of Zeroization |
|---|---|---|---|
| SSH Server Host Keys | The SSH server host keys to identify ssh server | Non-volatile storage/file system | Overwrite with zeros to clear cache and read verify, then erase file |
| SSH session keys | Keys exchanged for protecting the confidentiality of the remote administration session | Volatile storage | Openssh package is used but all keys are overwritten with zeros before freeing memory |
| SSH PKA | Public key authentication for remote administration over SSH | Non-volatile storage | Overwrite with zeros to clear cache and read verify, then erase file |
| X509 certificate with keys | For TLS connections | Non-volatile storage/file system | Overwrite with zeros to clear cache and read verify, then erase file |
| Local user password | User login | Non-volatile storage/file system (shadow file) | Erase file. Password is hashed with sha512 and the password file is only readable by root |
| TLS session HMAC keys | For TLS connections with Audit server | Volatile storage | OpenSSL package is used as infrastructure but all keys are overwritten with zeros before freeing memory |
| MACsec SAK | For securing MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |

| | | | |
|---|---|---|---|
| MACsec CAK | For deriving the SAK in MACsec Connections | Non-volatile storage/file system | Overwritten with zeros when deleted by administrator command. |
| MACsec Key Encryption Key (KEK) | For securing transport of the SAK in MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |
| MACsec Integrity Check Key (ICK) | For verifying the integrity of data in MACsec Connections | Volatile storage | Automatically when MACsec session terminated.<br>The value is zeroized by overwriting with another key or freed when the session expires. |

**Table 6 Keys and CSPs**.

The TOE can only be accessed through the CLI which implements the complete management interface of the TOE. The CLI does not implement any functions for displaying the symmetric keys, asymmetric private keys, passwords, or any other secret parameters

**NDcPP22e:FPT_STM_EXT.1**

The TOE implements a hardware clock for local date and time. The clock may be configured to use a locally configured time. The time is used for producing time stamps which are attached to audit records and to check the X.509 certificate expiration. The TOE also uses the clock to implement the session time out timers for each interactive session and to terminate each interactive session which exceeds the maximum allowed inactivity time.

**NDcPP22e:FPT_TST_EXT.1**

The TSF runs the following self-tests during initial start-up (on power on):

- Check of flash access and content with CRC (i.e, integrity check),
- Check of various Field-programmable gate array (FPGA) devices access and sanity
  - Verify control FPGA by writing a known value to a scratchpad area and verify it can be read back
- Probe the PCI bus and verify the devices are present as expected for that board type,
- Sanity check of memory to ensure no corruption
  - Error correction code (ECC) memory uncorrectable error verification, and check the appropriate memory size is reported by the driver for the board type
- Check of FANS for operational state
  - Verify fans presence and fans are not stalled on the 3926 devices via status information queried from the controlling FPGA
  - Verify power supply voltage and current operating values are within specification as queried from controlling FPGA
- Crypto KAT/self-test (including AES, SHS, HMAC, RSA, ECDSA and DRBG)

The performing of the above tests at startup are sufficient to demonstrate that the TOE is functioning properly.

**NDcPP22e:FPT_TUD_EXT.1**

The TOE implements a CLI command *show software* for querying the currently executing version of the TOE firmware and previous version if applicable. From time to time, the vendor makes available software upgrades at the product web site. The TOE allows the Security Administrators to manually upgrade the TOE software to the version available at the vendor's web site. Associated to each software upgrade, the vendor publishes a SHA-256 message digest value computed from the software upgrade. The user may compare the locally computed message digest to the message digest published at the vendor's web site to assert the authenticity of the upgrade. A new image is verified by the Administrator and then uploaded to the TOE where it is automatically installed.

The Security Administrator can obtain the software upgrade from the Ciena website and place it on a trusted file server. The TOE may then be instructed to connect to the file server and install the software image. The CLI command for installing the software upgrade is

```
software install url <url> tls-service-profile <tls-service-profile name>
```

A .sha256 file is a message digest is provided by the vendor as a separate download to help ensure the integrity of the provided image. The administrator can calculate the hash of the update image off box prior to install and match it against the .sha256 hash file to confirm the image is valid. If the calculated hash does not match the provided message digest, the administrator should not proceed with the installation. If it matches, the administrator can login and then proceed with the update. The installation process happens in one go as the reboot happens automatically, thus delayed activation is not supported, although the old version is retained on a separate partition and can be reverted back if necessary.

## 6.6 TOE access

**NDcPP22e:FTA_SSL.3**

The TOE will terminate a remote interactive session after a configurable time interval of session inactivity. The maximum allowed inactivity may be configured by the administrator of the TOE.

If a local or remote administrative session is inactive for a configured maximum period of inactivity, the session will be terminated. Fresh identification and authentication shall be required for the creation of a new session. The session inactivity timer will be restored for the new session

**NDcPP22e:FTA_SSL.4**

The TOE allows Administrators to terminate their own interactive sessions with the TOE using the `exit` command.

**NDcPP22e:FTA_SSL_EXT.1**

The TOE will terminate a local interactive session after a configurable time interval of session inactivity. If a local user session is inactive for a configured maximum period of inactivity, the TOE will terminate the session.

**NDcPP22e:FTA_TAB.1**

The TOE implements an administrator-configurable access banner which is displayed at each login window. Both methods of accessing the TOE (locally from console and remotely over SSH) require user authentication. The access banner is displayed at each login prompt.

## 6.7 Trusted path/channels

**NDcPP22e:FTP_ITC.1**

The TOE implements a TLS Client for a trusted channel between itself and authorized IT entities. The remote entity is an audit server. Each TLS connection is logically distinct from other communication channels and protects the channel data from disclosure and allows detection of the modification of the channel data. Peer entity authentication is with X.509 certificates for assured identification of the end points of the trusted channels.

**MACSEC10:FTP_ITC.1/MACSEC**

The TOE can be configured to establish MACsec connections with MACsec capable peers

**NDcPP22e:FTP_TRP.1/Admin**

The TOE implements a SSH server which allows SSHv2 connection between a remote management station and the TOE. A CLI which implements the management interface of the TOE is available to a remote administration over an encrypted SSHv2 channel.  The remote users (remote administrator) must initiate connection to the TOE using the SSH Client of the remote management station