Sierra Nevada Corporation

Binary Armor SCADA Network Guard, with firmware version 2.1

Security Target

Version 1.0

2021-06-04

Prepared for:

11551 East Arapahoe Road
Centennial, CO 80112

Prepared by:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Revision History		
Date	**Author**	**Modifications**
2020-05-11	Leidos	Initial draft
2020-05-18	Leidos	Updates to reflect product version 2.0
2020-06-16	Leidos	Updates to reflect vendor responses to outstanding issues
2020-09-24	Leidos	Updates to reflect vendor feedback
2020-10-05	Leidos	Updates to reflect evaluator feedback
2021-06-04	Leidos	Updates to reflect validator feedback, product version 2.1, and updated NIAP Technical Decisions

Table of Contents

List of Figures and Tables

1 Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE references, TOE overview, and TOE description. It also contains the ST and TOE conformance claims, ST conventions, glossary, and list of abbreviations.

This ST includes the following additional sections:

- Security Problem Definition (Section 2)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 3)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 4)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 5)—describes the security functions of the TOE and how they satisfy the SFRs
- Protection Profile Claims (Section 6)—provides rationale supporting the claims for conformance of the ST and the TOE to [cPPND]
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, Target of Evaluation, and Common Criteria Identification

ST Title: Sierra Nevada Corporation Binary Armor SCADA Network Guard, with firmware version 2.1 Security Target

ST Version: Version 1.0

ST Date: 2021-06-04

TOE Identification: Binary Armor SCADA Network Guard, with firmware version 2.1, consisting of:

- Binary Armor Hardware version 7000-SNC-01
- Binary Armor Firmware version 2.1

TOE Developer: Sierra Nevada Corporation

Evaluation Sponsor: Sierra Nevada Corporation

CC Identification: Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.

1.2 Conformance Claims

This ST and the TOE it describes are 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.

This ST and the TOE it describes are conformant to the following Protection Profile:

- *collaborative Protection Profile for Network Devices*, Version 2.2e, 23 March 2020 [cPPND], including the following optional and selection-based SFRs: FAU_STG.1, FAU_STG_EXT.3/LocSpace; FCS_HTTPS_EXT.1; FCS_NTP_EXT.1; FCS_TLSC_EXT.1; FCS_TLSC_EXT.2; FCS_TLSS_EXT.1; FCS_TLSS_EXT.2; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MTD.1/CryptoKeys; and FMT_MOF.1/Functions.

The following NIAP Technical Decisions are applicable to the claimed Protection Profile:

- TD0527 – Updates to Certificate Revocation Testing (FIA_X509_EXT.1)
  - This TD is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.
- TD0528 – NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4
  - This TD is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.
- TD0536 – NIT Technical Decision for Update Verification Inconsistency
  - This TD is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.
- TD0537 – NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3
  - This TD is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.
- TD0538 – NIT Technical Decision for Outdated link to allowed-with list
  - This TD is a semantic issue with the claimed PP that was corrected. It does not affect the ST or the evaluation of the TOE.
- TD0546 – NIT Technical Decision for DTLS - clarification of Application Note 63
  - This TD is not applicable to the TOE; it relates to FCS_DTLSC_EXT.1, which is not claimed by the ST.
- TD0547 – NIT Technical Decision for Clarification on developer disclosure of AVA_VAN
  - This TD is applicable to the TOE.
- TD0555 – NIT Technical Decision for RFC Reference incorrect in TLSS Test
  - This TD is applicable to the TOE but relates entirely to test evaluation activities so it does not affect the ST.
- TD0556 – NIT Technical Decision for RFC 5077 question
  - This TD is applicable to the TOE but relates entirely to test evaluation activities so it does not affect the ST.
- TD0563 – NIT Technical Decision for Clarification of audit date information
  - This TD is applicable to the TOE but relates entirely to an application note that clarifies the intent of FAU_GEN.1.2. It does not affect the ST.

- TD0564 – NIT Technical Decision for Vulnerability Analysis Search Criteria
  - This requirement is applicable to the TOE but relates entirely to the evaluation of AVA_VAN.1. It does not affect the ST because the additional ST evidence required to address this TD is already addressed by TD0547.
- TD0569 – NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7
  - This TD is applicable to the TOE but relates entirely to test evaluation activities so it does not affect the ST. Note specifically that while the title of the TD references FCS_DTLSS_EXT.1, which the ST does not claim, the TD also affects FCS_TLSS_EXT.1, which is within the TOE's logical boundary.
- TD0570 – NIT Technical Decision for Clarification about FIA_AFL.1
  - This TD is applicable to the TOE but relates entirely to clarifying how the PP reader should interpret FIA_AFL.1. It does not affect the ST.
- TD0571 – NIT Technical Decision for Guidance on how to handle FIA_AFL.1
  - This TD is applicable to the TOE but relates entirely to clarifying how the PP reader should interpret the requirements that relate to FIA_AFL.1. It does not affect the ST.
- TD0572 – NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers
  - This TD is applicable to the TOE but relates entirely to guidance on the interpretation of mandatory and allowable reference identifiers for cases where the TSF must validate the identifier of a presented X.509 certificate. It does not affect the ST.
- TD0580 – NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e
  - This TD is not applicable to the TOE. The TD affects a selection in FCS_CKM.2 that the ST does not claim.
- TD0581 – NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3
  - This TD is not applicable to the TOE. The TD adds a selection to FCS_CKM.2 that the ST does not claim.
- TD0591 – NIT Technical Decision for Virtual TOEs and hypervisors
  - This TD is not applicable to the TOE. The TD applies only to virtual network devices and there is no virtualized instance of the TOE.
- TD0592 – NIT Technical Decision for Local Storage of Audit Records
  - This TD is not applicable to the TOE. The TD modifies introductory text to remove a contradictory statement about local audit storage. It does not affect any SFR claims.

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; selection; assignment; and refinement.
  - Iteration—allows a component to be used more than once with varying operations. In this ST, the only iterated requirements are those reproduced from [cPPND], which uses descriptive strings to distinguish iterations of a requirement. For example, iterations of

FCS_COP.1 are identified FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.

- Selection—allows the specification of one or more elements from a list. Selections completed in the ST are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
- Assignment—allows the specification of an identified parameter. Assignments completed in the ST are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). An assignment within a selection is identified in bold italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Refinement—allows the addition of details. Refinements made in the ST of requirements drawn from [cPPND] would be indicated using bold for additions and strike-through for deletions (e.g., "… ~~some~~ **all** objects).

- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

1.4 Abbreviations and Acronyms

Table 1: Abbreviations and Acronyms

Abbreviation	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	Hash-based Message Authentication Code
ICS	Industrial Control System
MAC	Message Authentication Code
NSS	Network Security Services
NTP	Network Time Protocol
SCADA	Supervisory Control and Data Acquisition
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
USB	Universal Serial Bus

1.5 TOE Overview

The TOE is Binary Armor SCADA Network Guard, with firmware version 2.1 from Sierra Nevada Corporation (SNC). The product is a network device that is used to interface with Supervisory Control and Data Acquisition (SCADA) network systems for real-time monitoring purposes.

The focus of this evaluation is on the TOE functionality supporting the claims in the collaborative Protection Profile for Network Devices ([cPPND] – see Section 1.2 for specific version information). The security functionality specified in [cPPND] includes protection of communications between the TOE and external IT entities, identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, and use of NIST-validated cryptographic mechanisms. In particular, [cPPND] does not define requirements for interaction with SCADA systems so these interfaces are not included in the TOE boundary.

1.6 TOE Description

The TOE is intended for in-line installation between Programmable Logic Controllers (PLCs), remote terminal units, intelligent electronic devices or controllers and the WAN/LAN, to provide bi-directional security across all communication layers. It provides two, separate, physical interfaces: a “high” network interface card (NIC), typically connected to SCADA/ICS equipment; and a “low” NIC, typically connected to external systems such as Human Machine Interface. The TOE supports remote administration over the network (from either the high or low networks) and local administration through a directly networked workstation.

The product works by processing every byte of every message with a dynamic state-based rule-set that processes messages based on system control logic. This process ensures only safe message traffic reaches critical SCADA systems. The product’s ability to validate ICS communications is outside the scope of the TOE as there are no Protection Profile claims for ICS protocols.

For the purpose of this evaluation, the TOE is treated as a network device offering CAVP certified cryptographic functions, security auditing, secure administration, trusted updates, self-tests, and secure connections to other servers (e.g., to export audit records), protected using TLS.

1.6.1 Physical Scope

The TOE is provided as a hardware network appliance. There are no virtual deployments of the TOE and the TOE is always a standalone device; its functionality is not distributed across multiple devices.

The TOE has a rugged enclosure that protects it from modification and contains a single embedded board containing an Intel Atom E3845 processor, memory, and flash storage. The TOE hardware contains a hardened operating system (RHEL 7) that does not permit operators (even an authorized administrator) access to the OS, with SNC-developed firmware running atop. The TOE provides a TLS-protected management interface that can be accessed via SNC’s Administration Tool application. This runs on a PC/workstation that is part of the operational environment of the TOE. This application is a graphical front-end for interacting with the TSF through a REST API; the REST API may also be invoked directly to interact with the TOE and its data. An administrator can configure the TOE for remote access on either its high or low network interface. The administrator always accesses the TOE through its TLS management interface, irrespective of whether the administrator configured the TOE to listen for management connections on its low or high network interface and irrespective of whether the administrator accesses the TOE remotely or locally. In this context, local access means connected via crossover cable or through a network switch to which only the TOE and the workstation are connected.

The TOE can be configured to forward its audit records to an external syslog server in the network environment. This is generally advisable given the audit log storage space on the evaluated appliance.

The TOE in its evaluated configuration requires the following components in its operational environment:

- A Microsoft Windows 10 or Redhat Linux version 7.7+ workstation—with the Binary Armor software suite of tools installed.
- A security token in the form of a PKCS#11-compliant smart card or USB device present on the administrative workstation. The security token is used by the TOE for administrator authentication. The token is configured by loading public key(s) onto the TOE.[1]
- A TLS-protected syslog server that receives audit events from the TOE
- An NTP server with which the TOE can synchronize its clock.

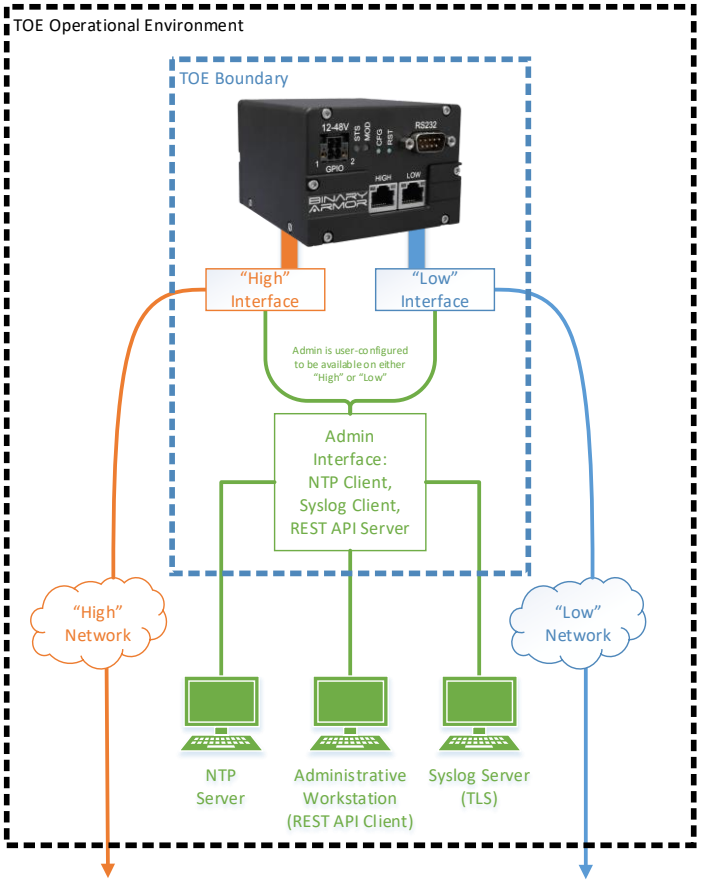
External servers are expected to be connected to the low network by default, but the TSF can be configured to interact with these servers on the high network as well if required.

In addition to the TOE firmware itself, the TOE contains the following licensed third-party components that will be considered during the vulnerability analysis:

- Red Hat Enterprise Linux, version 7.9
- Boost C++ libraries, version 1.70.1
- Qt framework (unversioned)
- Roboto font family (unversioned)
- OpenSSL, version 1.0.2k
- OpenSC PKCS#11 wrapper library, version 0.19.0
- Xerces XML library, version 3.1.3
- Lua, version 5.3.5

The TOE boundary excludes SCADA interfaces and the Enterprise Client Management Tool. These interfaces are present on the product but are non-interfering with respect to security as their presence does not prevent any of the TOE's SFR claims from being satisfied when operated in its evaluated configuration. The TOE boundary also excludes the physical chassis buttons used for resetting the device, 'override' mode used to define SCADA rulesets, and management of what are referred to as 'configuration files'. The physical chassis buttons and override mode do not relate to any claimed security functionality, and the configuration files do not affect any security-relevant data with respect to the PP claims. Use of these interfaces is prevented through the achievement of the OE.PHYSICAL environmental objective. Finally, the TOE boundary excludes the RS-232 serial interface depicted on the figure below. It is physically part of the product but only used for a local interface to certain SCADA devices; all communications over this interface are therefore physically protected and its use is not security-relevant.

[1] The security token related functions (pairing, signing, encrypting and activating) have not been evaluated and are outside the scope of this evaluation. Password-based authentication is also supported.



1.6.2 Logical Scope

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.

1.6.2.1 Security Audit

The TOE generates audit events associated with identification and authentication, management, updates, and user sessions. The TOE can store the events in a local log or export them to a syslog server using a TLS protected channel.

1.6.2.2 Cryptographic Support

The TOE provides CAVP certified cryptography in support of its TLS implementation and administrator authentication. Cryptographic services include key management, random bit generation, symmetric encryption and decryption, digital signature, and secure hashing.

1.6.2.3 Identification and Authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of reading the login banner, obtaining status, and requesting the TOE's public key certificate. It provides the ability to both assign credentials (user password, enrollment for PKCS#11 token) and to authenticate users against these credentials. The TOE also provides X.509 certificate checking for its TLS connections. The password-based authentication will cause a lockout in the event of an excessive number of consecutive authentication failures.

1.6.2.4 Security Management

The TOE provides a management interface that an administrator can access via a network port. The SNC Administration Tool application uses the TOE's REST API. This API can also be accessed directly over HTTPS. The management interface is protected with TLS. The management interface is limited to the authorized administrator.

1.6.2.5 Protection of the TSF

The TOE implements various self-protection mechanisms. The TOE performs self-tests that cover the correct operation of the TOE. It provides functions necessary to securely update the TOE. It relies upon either manually provided time or an NTP server in its environment to ensure reliable timestamps. It protects sensitive data such as passwords and cryptographic keys stored on the TOE's internal Flash so that they are not accessible even by an authorized administrator.

1.6.2.6 TOE Access

The TOE will terminate local and remote interactive sessions after a configurable period of inactivity. The TOE additionally provides the capability for administrators to terminate their own interactive sessions. The TOE can be configured to display an advisory and consent warning message before establishing a user session.

1.6.2.7 Trusted Path/Channels

The TOE provides local administration which is subject to physical protection. To access the TOE locally, an operator must directly network their workstation to the TOE (again, e.g., a crossover cable or through a network switch to which only the TOE and the workstation are connected). For both local and remote access, the administrative session is protected by TLS, thus ensuring protection against modification and disclosure.

The TOE also protects exported audit records from modification and disclosure by using TLS to communicate with the syslog server.

1.7 TOE Documentation

The TOE is supplied with the following guidance documentation that describes the installation process for the TOE and provides guidance for configuration and secure use of its security features:

- ADMINISTRATION GUIDE for COMMON CRITERIA for Binary Armor® SCADA Network Guard, with Firmware version 2.1, 0318-0200-0004 Document Number Rev A (CC Admin Guide)
- BINARY ARMOR® USER MANUAL 0318-0100-0015 Document Number Rev G (User Manual)

- Binary Armor SCADA Network Guard, with Firmware version 2.1, Binary Armor Alloy API, version 0.3 (API).

2 Security Problem Definition

This ST includes by reference the Security Problem Definition (comprising threat statements, assumptions, and organizational security policies) from [cPPND]. The PP offers additional information about the threats, assumptions, and organizational security policies, but that has not been reproduced here and the PP should be consulted if there is interest in that material.

In general, the [cPPND] has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the TOE.

The following elements of the SPD from [cPPND] are not applicable to the TOE:

- A.COMPONENTS_RUNNING – applies to distributed TOEs only; the TOE is not distributed.
- A.VS_TRUSTED_ADMINISTRATOR – applies to virtual network devices only; the TOE does not have a virtualized deployment.
- A.VS_REGULAR_UPDATES – applies to virtual network devices only; the TOE does not have a virtualized deployment.
- A.VS_ISOLATION – applies to virtual network devices only; the TOE does not have a virtualized deployment.
- A.VS_CORRECT_CONFIGURATION – applies to virtual network devices only; the TOE does not have a virtualized deployment.

This is acceptable because [cPPND] specifically states that these assumptions are conditional on TOE deployments that not all valid TOEs will have.

3 Security Objectives

The [cPPND] defines the following security objectives for the operational environment of the TOE.

Table 2: Security Objectives for the Operational Environment

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

The following environmental security objectives from [cPPND] are not applicable to the TOE:

- OE.COMPONENTS_RUNNING – applies to distributed TOEs only; the TOE is not distributed.
- OE.VM_CONFIGURATION – applies to virtual network devices only; the TOE does not have a virtualized deployment.

This is acceptable because [cPPND] specifically states that these objectives are conditional on TOE deployments that not all valid TOEs will have.

4 IT 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 TOE and to scope the evaluation effort.

The SFRs have all been drawn from [cPPND]. As such, operations on SFRs already performed in that PP are not identified here. Rather, the SFRs have been copied from [cPPND] and any formatting used in that PP has been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs are the set of SARs specified in [cPPND].

4.1 Extended Requirements

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

- FAU_STG_EXT.1—Protected Audit Event Storage
- FAU_STG_EXT.3/LocSpace Action in Case of Possible Audit Data Loss
- FCS_HTTPS_EXT.1 – HTTPS Protocol
- FCS_NTP_EXT.1—NTP Protocol
- FCS_RBG_EXT.1—Random Bit Generation
- FCS_TLSC_EXT.1—TLS Client Protocol
- FCS_TLSC_EXT.2 – TLS Client Support for Mutual Authentication
- FCS_TLSS_EXT.1—TLS Server Protocol
- FCS_TLSS_EXT.2 – TLS Server Support for Mutual Authentication
- FIA_PMG_EXT.1—Password Management
- FIA_UAU_EXT.2—Password-Based Authentication Mechanism
- FIA_UIA_EXT.1—User Identification and Authentication
- FIA_X509_EXT.1—X.509 Certificate Validation
- FIA_X509_EXT.2—X.509 Certificate Authentication
- FIA_X509_EXT.3—X.509 Certificate Requests
- FPT_APW_EXT.1—Protection of Administrator Passwords
- FPT_SKP_EXT.1—Protection of TSF Data
- FPT_STM_EXT.1—Reliable Time Stamps
- FPT_TST_EXT.1—TSF Testing
- FPT_TUD_EXT.1—Trusted Update
- FTA_SSL_EXT.1—TSF-Initiated Session Locking

4.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 3: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1—Audit Data Generation
	FAU_GEN.2—User Identity Association
	FAU_STG.1 – Protected Audit Trail Storage

Requirement Class	Requirement Component
	FAU_STG_EXT.1—Protected Audit Event Storage
	FAU_STG_EXT.3/LocSpace – Action in Case of Possible Audit Data Loss
FCS: Cryptographic support	FCS_CKM.1—Cryptographic Key Generation
	FCS_CKM.2—Cryptographic Key Establishment
	FCS_CKM.4—Cryptographic Key Destruction
	FCS_COP.1/DataEncryption—Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen—Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash—Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash—Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1 – HTTPS Protocol
	FCS_NTP_EXT.1—NTP Protocol
	FCS_RBG_EXT.1—Random Bit Generation
	FCS_TLSC_EXT.1—TLS Client Protocol without Mutual Authentication
	FCS_TLSC_EXT.2 – TLS Client Support for Mutual Authentication
	FCS_TLSS_EXT.1—TLS Server Protocol without Mutual Authentication
	FCS_TLSS_EXT.2 – TLS Server Support for Mutual Authentication
FIA: Identification and authentication	FIA_AFL.1—Authentication Failure Management
	FIA_PMG_EXT.1—Password Management
	FIA_UAU_EXT.2—Password-Based Authentication Mechanism
	FIA_UAU.7—Protected Authentication Feedback
	FIA_UIA_EXT.1—User Identification and Authentication
	FIA_X509_EXT.1/Rev—X.509 Certificate Validation
	FIA_X509_EXT.2—X.509 Certificate Authentication
	FIA_X509_EXT.3—X.509 Certificate Requests
FMT: Security management	FMT_MOF.1/Functions—Management of Security Functions Behavior
	FMT_MOF.1/ManualUpdate—Management of Security Functions Behavior
	FMT_MTD.1/CoreData—Management of TSF Data
	FMT_MTD.1/CryptoKeys—Management of TSF Data
	FMT_SMF.1—Specification of Management Functions
	FMT_SMR.2—Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1—Protection of Administrator Passwords
	FPT_SKP_EXT.1—Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	FPT_STM_EXT.1—Reliable Time Stamps
	FPT_TST_EXT.1—TSF Testing

Requirement Class	Requirement Component
	FPT_TUD_EXT.1—Trusted update
FTA: TOE access	FTA_SSL_EXT.1—TSF-Initiated Session Locking
	FTA_SSL.3—TSF-Initiated Termination
	FTA_SSL.4—User-Initiated Termination
	FTA_TAB.1—Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1—Inter-TSF Trusted Channel
	FTP_TRP.1/Admin—Trusted Path

4.2.1 Security Audit (FAU)

4.2.1.1 Audit Data Generation (FAU_GEN.1)

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 4.

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 4.

Table 4: Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.3/LocSpace	Low storage space for audit events	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session	Reason for failure
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	Identity of new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_TLSC_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSC_EXT.2	None.	None.
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.2	Failure to authenticate the client	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1).	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	None.

4.2.1.2 User Identity Association (FAU_GEN.2)

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.

4.2.1.3 Protected Audit Trail Storage (FAU_STG.1)

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

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

4.2.1.4 Protected Audit Event Storage (FAU_STG_EXT.1)

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

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

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

FAU_STG_EXT.1.3 The TSF shall [***overwrite previous audit records according to the following rule: [overwriting the oldest log record first]***] when the local storage space for audit data is full.

4.2.1.5 Action in Case of Possible Audit Data Loss (FAU_STG_EXT.3/LocSpace)

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

4.2.2 Cryptographic Support (FCS)

4.2.2.1 Cryptographic Key Generation (FCS_CKM.1)

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 cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1***

].

4.2.2.2 Cryptographic Key Establishment (FCS_CKM.2)

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 meets the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***

].

4.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

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.

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC***, ***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***, ***GCM as specified in ISO 19772***].

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

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, 3072 bits]***
- ***Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 bits, 521 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***

].

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

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.

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

FCS_COP.1.1/KeyedHash The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1***, ***HMAC-SHA-256***, ***HMAC-SHA-384***] and cryptographic key sizes [**512 bits for HMAC-SHA-1 and HMAC-SHA-256, 1024 bits for HMAC-SHA-384**] and message digest sizes [***160***, ***256***, ***384***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

4.2.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [***not establish the connection***] if the peer certificate is deemed invalid.

Application Note: *This SFR is only applicable in the case where the TOE is configured to use mutual authentication. When mutual authentication is not used, a peer certificate is not requested by the TSF because the TOE only implements HTTPS as a server. When mutual authentication is used, the 'not establish the connection' selection applies, consistent with FCS_TLSS_EXT.2.2.*

4.2.2.9 NTP Protocol (FCS_NTP_EXT.1)

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [***NTP v3 (RFC 1305), NTP v4 (RFC 5905)***].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

- ***Authentication using [SHA256, SHA384, SHA512] as the message digest algorithm(s)];***

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

4.2.2.10 Random Bit Generation (FCS_RBG_EXT.1)

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

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

Application Note: *The TOE's DRBG is seeded with a minimum entropy of 384 bits, but this is not a selection offered by the claimed PP. Given that this will always exceed 256 bits, "256 bits" has been selected and the intent of the requirement is met.*

4.2.2.11 TLS Client Protocol without Mutual Authentication (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] 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_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***

- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

].

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 CN or SAN***].

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***

].

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.

4.2.2.12 TLS Client Support for Mutual Authentication (FCS_TLSC_EXT.2)

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

4.2.2.13 TLS Server Protocol without Mutual Authentication (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] 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_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***

- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0.

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [***RSA with key size [2048 bits, 3072 bits]***, ***Diffie-Hellman parameters with size [2048 bits], ECDHE curves [secp256r1] and no other curves***].

FCS_TLSS_EXT.1.4 The TSF shall support [***no session resumption or session tickets***].

4.2.2.14 TLS Server Support for Mutual Authentication (FCS_TLSS_EXT.2)

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

FCS_TLSS_EXT.2.2 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- ***Not implement any administrator override mechanism***

].

FCS_TLSS_EXT.2.3 The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

4.2.3 Identification and Authentication (FIA)

4.2.3.1 Authentication Failure Management (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [**3..10**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

4.2.3.2 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")"***];
b) Minimum password length shall be configurable to between [**4**] and [**15**] characters.

4.2.3.3 Password-Based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1 The TSF shall provide a local [***password-based, certificate-based***] authentication mechanism to perform local administrative user authentication.

4.2.3.4 Protected Authentication Feedback (FIA_UAU.7)

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.

4.2.3.5 User Identification and Authentication (FIA_UIA_EXT.1)

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;
- [***[Obtain status information,***
- ***Request TOE certificate]***].

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.

4.2.3.6 X.509 Certificate Validation (FIA_X509_EXT.1/Rev)

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

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The 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 [***a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

4.2.3.7 X.509 Certificate Authentication (FIA_X509_EXT.2)

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

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

4.2.3.8 X.509 Certificate Requests (FIA_X509_EXT.3)

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

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

4.2.4 Security Management (FMT)

4.2.4.1 Management of Security Functions Behavior (FMT_MOF.1/ManualUpdate)

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

4.2.4.2 Management of Security Functions Behavior (FMT_MOF.1/Functions)

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

4.2.4.3 Management of TSF Data (FMT_MTD.1/CoreData)

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

4.2.4.4 Management of TSF Data (FMT_MTD.1/CryptoKeys)

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

4.2.4.5 Specification of Management Functions (FMT_SMF.1)

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 [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
  - ***Ability to modify the behavior of the transmission of audit data to an external IT entity*;**
  - ***Ability to manage the cryptographic keys*;**
  - ***Ability to set the time which is used for time-stamps*;**
  - ***Ability to configure NTP*;**
  - ***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 stor*e;**

].

4.2.4.6 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

4.2.5 Protection of the TSF (FPT)

4.2.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

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

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

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

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

4.2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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

FPT_STM_EXT.1.2 The TSF shall [***allow the Security Administrator to set the time*, *synchronise time with an NTP server***].

4.2.5.4 TSF Testing (FPT_TST_EXT.1)

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: [**OpenSSL performs self-test including the following cryptographic algorithm Known Answer Tests (KATs): AES, RSA, ECDSA, DH, ECDH, DRBG, HMAC, and SHA and an integrity test: HMAC-SHA-256. NSS performs the follows KATs: ECDSA, RSA, and SHA and a module integrity test using DSA 2048 w/ SHA-256; the TOE decrypts stored data at rest using AES**].

4.2.5.5 Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [***no other TOE firmware/software version***].

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

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

4.2.6 TOE Access (FTA)

4.2.6.1 TSF-Initiated Session Locking (FTA_SSL_EXT.1)

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

4.2.6.2 TSF-Initiated Termination (FTA_SSL.3)

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

4.2.6.3 User-Initiated Termination (FTA_SSL.4)

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

4.2.6.4 Default TOE Access Banners (FTA_TAB.1)

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.

4.2.7 Trusted Path/Channels (FTP)

4.2.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

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

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

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

4.2.7.2 Trusted Path (FTP_TRP.1/Admin)

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

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

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

4.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the [cPPND].

Table 5: Assurance Components

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

5 TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 4.2:

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

5.1 Security Audit

5.1.1 Audit Data Generation

The TOE provides an audit log (the ba.log) and provides logging of the audit information described in the table in section 4.2.1.1. All auditable events that involve configuration of the TSF include the action requested, the success or failure, and the source IP address of the request. This includes any cryptographic operations, specifically generation of a CSR, import of a signed response, or import of a certificate. When auditing all administrative interactions with key data (i.e. certificate operations and certificate-based authentication attempts), the TOE records the certificate's Subject, Issuer, Valid dates, and Key Type in order to uniquely identify the certificate. When an administrator is authenticated to the TOE, subsequent events following authentication identify the administrator using the certificate Subject field.

This aspect of the Security Audit security function satisfies FAU_GEN.1 and FAU_GEN.2.

5.1.2 Audit Storage and Audit Record Export

The TOE is a single standalone appliance that stores audit logs locally and provides the administrator the ability to configure the real-time export of syslog records protected with TLS. The TOE provides 100 megabytes of local storage for audit logs, and the TOE permits no access to the audit logs other than allowing an authenticated administrator to download a copy of the logs or to clear the logs. The TOE will delete the oldest audit records in order to free up space for new audit records when the TOE reaches the maximum local storage space for audit data. The TOE provides ten rotations and ensures that the oldest rotation is deleted. Audit records are protected against unauthorized access and can only be cleared by a Security Administrator. When a rotation occurs, a log event is generated.

This aspect of the Security Audit security function satisfies FAU_STG.1, FAU_STG_EXT.1, and FAU_STG_EXT.3/LocSpace.

5.2 Cryptographic Support

The TOE includes the Red Hat OpenSSL and Red Hat Network Security Services (NSS) libraries. Each of these libraries possess CAVP certificates for their different cryptographic algorithms. Tables 6 and 7 below summarize the CAVP certificates.

The TOE uses its Red Hat OpenSSL library for all TLS and certificate functionality.

Table 6: Cryptographic Functions Implemented by OpenSSL

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048, 3072 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	RSA #2976
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	ECDSA #1495
DSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	DSA #1425
Key establishment (FCS_CKM.2)		
RSA-based scheme	RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447	N/A
Elliptic curve-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Component #1986
Finite field-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	#A1150 (parameter set FB and FC)
Data encryption (FCS_COP.1/DataEncryption)		
AES in CBC mode (128, 256 bits) AES in GCM mode (128, 256 bits)	ISO 18033-3 (AES) ISO 10116 (CBC mode) ISO 19772 (GCM mode)	AES #5544
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (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 RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	RSA #2976
ECDSA Elliptic Curve Digital Signature Algorithm (P-256, P-384, P-521 curves)	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	ECDSA #1495
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	SHS #4450

Functions	Standards	Certificates
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-1 (key size 512 bits, digest size 160 bits) HMAC-SHA-256 (key size 512 bits, digest size 256 bits) HMAC-SHA-384 (key size 1024 bits, digest size 384 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	HMAC #3695
Deterministic random bit generation (FCS_RBG_EXT.1)		
Counter DRBG	ISO/IEC 18031:2011	DRBG #2199

The TOE uses its Red Hat NSS library to verify the signatures of trusted updates (through RPM and the yum package manager).

Table 7: Cryptographic Functions Implemented by NSS

Functions	Standards	Certificates
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (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 RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	RSA #2975
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	SHS #4449

5.2.1 Cryptographic Operations

The TOE uses most cryptographic algorithms in support of TLS. For example, the TOE uses SHA hashing both during digital signature calculation (hashing of the message) as well as for integrity as part of HMAC-SHA operations within TLS. The TOE implements AES-CBC and GCM (both 128 and 256-bit) depending upon the TLS cipher suite. Likewise, during TLS authentication, the TOE will generate 2048-bit finite field Diffie-Hellman parameters, or ECDSA P-256/P-384/P-521 keys for key establishment depending on the negotiated ciphersuite, the cryptographic parameters supported by the remote endpoint of the connection, and whether the TOE is acting as the client or server in the connection. When an RSA cipher suite is used, the TOE uses its certificate public key for key establishment. This 2048-bit or 3072-bit public key is generated by the TOE when it generates a key pair as part of a certificate signing request. The TOE uses HMAC-SHA-1, HMAC-SHA-256, or HMAC-SHA-384 (depending on negotiated cipher suite) for integrity of TLS protected data using SHA-1/256/384 with 512/1024-bit keys to produce a 160/256/384 output MAC. The HMAC-SHA-1 and HMAC-SHA-256 algorithms have a block size of 512 bits, while the HMAC-SHA-384 algorithm has a block size of 1024 bits. The TOE also uses cryptography when verifying the signatures of updated packages and will use 3072-bit RSA for this.

This aspect of the Cryptographic Support security function satisfies FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.

5.2.2 Random Bit Generation

The TOE uses an AES-256 CTR_DRBG from OpenSSL library, which is seeded with a minimum of 384 bits of entropy drawn from `/dev/random` that is populated by a platform-based noise source. The TOE uses this DRBG to generate all keys (as part of TLS and CSR key generation) as well as to generate salts and nonces (for password hashing and TLS respectively).

This aspect of the Cryptographic Support security function satisfies FCS_RBG_EXT.1.

5.2.3 Cryptographic Key Generation and Establishment

The TOE supports generating key pairs, both for authentication and for key exchange. When generating authentication key pairs, the TOE can generate a CSR with an RSA 2048 or 3072 bit key pair and an ECDSA P-256 or P-384 key pair. For key establishment, the TOE will generate 2048-bit DHE keys, or P-256/P-384/P-521 ECDHE keys, or 2048/3072- bit RSA keys (depending on the negotiated cipher suite) during TLS negotiation. The TOE's TLS server only uses P-256 while the TLS client supports all P-256, P-384, and P-521.

The TOE performs key establishment for TLS, and the TOE acts as both a TLS server (to service incoming administrative sessions) and as a TLS client (for syslog export). The TOE supports RSA, elliptic curve, and finite field key establishment methods for TLS, depending on the negotiated cipher suite.

This aspect of the Cryptographic Support security function satisfies FCS_CKM.1 and FCS_CKM.2.

5.2.4 Cryptographic Key Destruction

The TOE clears keys from both volatile (by overwriting the memory locations in RAM with zeroes) and non-volatile (by first overwriting the contents of the file in the Flash filesystem with zeroes, then `sync`'ing, and finally deleting the file) memory.

Table 8: Key Clearing

Key/CSP	Storage Location	Zeroized upon:	Zeroized by:
TLS Host RSA or ECDSA private key	On Flash	Command	Overwriting with zeroes (file system APIs)
TLS pre-master secret	Volatile memory (cleartext)	Handshake done	Overwriting with zeroes (OpenSSL)
TLS session key	Volatile memory (cleartext)	Close of session	Overwriting with zeroes (OpenSSL)
Passwords	On Flash in a SHA-256 hash	Command	Overwriting with zeroes (file system APIs)

Keys stored on Flash are stored in an encrypted format; however, as the key storage locations are not accessible through any interface designed for that purpose (refer to FPT_SKM_EXT.1), the encryption method was not considered and the key storage is assumed to provide equivalent strength to cleartext.

This aspect of the Cryptographic Support security function satisfies FCS_CKM.4.

5.2.5 Cryptographic Protocols

The TOE implements the following cryptographic protocols to protect communications between itself and non-TOE entities:

- TLS as a client—the TOE acts as a TLS client when exporting audit records to an external audit server
- TLS as a server—the TOE acts as a TLS server supporting inbound administrative sessions

In both cases, mutual authentication is supported for TLS.

- NTP—the TOE can synchronize its system clock with an NTP server.

5.2.5.1 TLS Client Protocol

The TOE's TLS client supports TLS 1.1 and TLS 1.2 with the following TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The TOE allows the TLS client (used for audit export) to specify the syslog server by hostname (which uses the internal `hosts` file for resolution) or IPv4 address. The TOE's TLS client implementation establishes its reference identifiers from the administrator-configured reference identifiers per Section 6 of RFC 6125, using the hostname or IPv4 address (in CN or SAN) as a reference identifier and checking that the syslog server's certificate includes the specified identifier. The IP address is converted to a binary representation and examined per RFC 3986. The TOE supports handling of wildcards within certificates. The TLS client supports the Elliptic Curves Extension (specifying only P-256, P-384, and P-521) in its Client Hello, and the TOE does not require (nor allow) administrative configuration of this extension; the TOE always sends it. The TOE's TLS client is capable of providing a client certificate to the remote server in support of mutual authentication. If the TLS server certificate is invalid, the TSF will automatically reject it without any administrative override capability.

This aspect of the Cryptographic Support security function FCS_TLSC_EXT.1 and FCS_TLSC_EXT.2.

5.2.5.2 TLS Server Protocol

The TOE's TLS server supports TLS 1.1 and TLS 1.2 with the following TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The TOE denies versions of TLS older than TLS 1.1 and does not support session resumption or session tickets. The TOE uses 2048/3072-bit RSA, 2048-bit DHE, or an elliptic curve during TLS key exchange. Specifically, the TLS server uses P-256 while the TLS client can use any of P-256, P-384, or P-521. The TOE's TLS server is capable of requesting and validating a client certificate from the remote client in support of mutual authentication. When mutual authentication is enabled and the client certificate cannot be validated, the connection is rejected without any option for override. In this case, certificate validation is done for validity (e.g. cryptographically valid, not expired, not revoked) as well as for identification of the client. Specifically, the IPv4 address in the CN or SAN of the client certificate is validated against the expected value per RFC 6125 section 6, and the certificate is rejected if the reference identifier is invalid. If mutual authentication is not enabled, the client certificate will not be validated even if one is presented to the TOE.

The TOE implements TLS server functionality for inbound REST API requests over HTTPS. These may be initiated through direct invocation or indirectly through use of the Administration Tool application, which interfaces with the TSF through abstraction of REST API calls to a user interface. The TOE's HTTPS implementation conforms to RFC 2818.

This aspect of the Cryptographic Support security function satisfies FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, and FCS_TLSS_EXT.2.

5.2.5.3 NTP Protocol

The TOE can synchronize its system clock with an NTP server. The TOE supports both NTPv3 as defined in RFC 1305 and NTP v4 as defined in RFC 5905 and can use SHA-256, SHA-384, or SHA-512 as its means for authenticating the NTP timestamps it receives from configured NTP servers. The TOE can support up to three NTP time sources and will not update NTP timestamps from broadcast or multicast addresses.

This aspect of the Cryptographic Support security function satisfies FCS_NTP_EXT.1.

5.3 Identification and Authentication

5.3.1 User Identification and Authentication

The TOE requires that the administrator first authenticate through the TOE's TLS protected management interface before permitting any sensitive services. Administrator authentication is performed using a PKCS#11 token from a file, smart card, or other hardware authentication device and optionally a password. Certificate authentication is performed using the administrator's public key. Password authentication is performed using the administrator's certificate as their identity and the password is used as an authenticator for the claimed identity. In both cases, the validity of the authentication is determined locally within the TOE. Administrators enroll their PKCS#11 token with the TOE so that an external authentication server interface is not required. A valid password is determined through hash comparison. The password hash is associated with the user via the user certificate's DN. A valid PKCS#11 token is determined through the token bearer using its private key to decrypt a known value that was encrypted by the TOE using the bearer's public key and provided back to the TOE for comparison to the original value. Without authenticating, the TOE will permit a user to view the banner, query status information (network statistics, network metrics, output the hash of the current configuration), and query the TOE's certificate. As described in section 1.6.1, the administrator uses the same TLS management interface for all access (i.e., high and low interfaces, local and remote).

The TOE provides no feedback to the administrator during authentication other than the success or failure of their attempt. For password-based credentials, the TOE accepts passwords that consist of any combination of uppercase letters, lowercase letters, numbers, and the special characters !, @, #, $, %, ^, &, *, (, and). Passwords must meet a minimum required length that is administratively configurable to between 4 and 15 characters.

This aspect of the Identification and Authentication security function satisfies FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7, and FIA_PMG_EXT.1.

5.3.2 Authentication Failure Management

For password-based authentication, the administrator can configure the number of incorrect remote authentication attempts to a value between 3 and 10 (with the default being 3), and if one exceeds that threshold, the TOE will enforce an administrator-configurable (between 1 and 60 seconds) lockout of the administrator. Administrator accounts that authenticate solely with a public key are not subject to lockout.

This aspect of the Identification and Authentication security function satisfies FIA_AFL.1.

5.3.3 X.509 Certificate Validation

The TOE performs revocation checking of certificates during TLS client connection establishment (i.e., when the TOE acts as a TLS client connecting to a remote syslog-tls server) using CRLs that are identified by the crlDistributionPoints extension in presented certificates. It also performs revocation checking of a presented client certificate presented for remote administration. The TOE's CRL validation mechanism conforms to RFC 5280. The TOE performs the revocation checking after having checked the validity of the server certificate and its chain, conformant to RFC 5280. The TOE requires that TLS server certificates have the Server Authentication purpose and that TLS client certificates have the Client Authentication purpose in order to be considered valid. The TOE will not treat a CA certificate as such unless the basicConstraints

extension is present with the CA flag set to TRUE. The TOE also ensures that any TLS client and server certificates that are presented to it have the proper purpose identified in the extendedKeyUsage field. The TSF does not use certificates for OCSP, software updates, or self-tests of code integrity so the extendedKeyUsage field values that correspond to these uses are not applicable.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.1/Rev.

5.3.4 X.509 Certificate Authentication

The TOE uses X.509 certificates for authentication of TLS and HTTPS trusted channels. The TOE relies upon the administrator to load the CA certificates (root CA and any needed intermediate certificates). If the presented certificate's revocation status cannot be determined because the associated CRL cannot be accessed, the TOE treats the certificate as invalid and rejects it.

When establishing connectivity to a remote syslog server or receiving a connection from a remote administrator when mutual authentication is enabled, the TSF will ensure that the presented certificate chains to a trusted root CA in its trust store. When receiving a connection from a remote administrator and when using mutual authentication for syslog server connectivity, the TSF will present its own client or server certificate to the peer, depending on the connection.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.2.

5.3.5 X.509 Certificate Requests

The TOE allows an administrator to request the TOE perform on-board key generation of an RSA or ECDSA key pair and then outputs a Certificate Signing Request (CSR), which the administrator can have signed by a suitable CA.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.3.

5.4 Security Management

5.4.1 Security Roles and Specification of Management Functions

In the evaluated configuration, a single administrative role (the Security Administrator) is available to administer the TOE both locally and remotely. The TOE provides the Security Administrator administrative access through its HTTPS server. The TOE is accessed remotely by accessing the running HTTPS server. The TOE is accessed locally by connecting a workstation directly to a network port via crossover cable or local switch.

The TOE provides the following management functions:

- Configure the access banner
- Configure the session inactivity time before session termination
- Update the TOE and verify TOE updates prior to installation using digital signature verification
- Configure the authentication failure parameters
- Modify the behavior of the transmission of audit data to an external syslog server by configuring the target
- Manage cryptographic keys
- Set the time used for time stamps
- Configure NTP

- Manage the TOE's trust store and designate X509 v3 certificates as trust anchors
- Import X.509 v3 certificates to the TOE's trust store.

All management functions are implemented through the REST API. This can be invoked directly by an administrator or indirectly through the SNC Administration Tool application which provides a graphical front-end for the API used to configure the TOE. The management functionality available to the administrator does not differ based on whether the TOE is accessed locally or remotely.

This aspect of the Security Management security function satisfies FMT_SMR.2 and FMT_SMF.1.

5.4.2 Management of Security Functions Behavior

The ability to modify the behavior of transmission of audit data to an external IT entity and to perform TOE updates is restricted to the single administrative role of Security Administrator. The administrator can modify the values for the syslog server connection using the Administration tab of the Administration Tool.

This aspect of the Security Management security function satisfies FMT_MOF.1/Functions and FMT_MOF.1/ManualUpdate.

5.4.3 Management of TSF Data

The TOE provides only non-security relevant status information to Security Administrators prior to authentication and restricts the ability to manage cryptographic keys to Security Administrators. A user may only access the TOE as a Security Administrator if they are defined as such within the TOE and associated with a public key associated with their token. Specifically, the Security Administrator may use the TOE to generate CSRs (which contain key pairs), load public keys for administrator tokens, and load certificates (whether it is certificate for the TOE generated by an external CA or a certificate used to validate a presented TLS client or server certificate). This functionality is performed using the Administration Tool or directly via invocation of the REST API.

This aspect of the Security Management security function satisfies FMT_MTD.1/CoreData and FMT_MTD.1/CryptoKeys.

5.5 Protection of the TSF

5.5.1 Protection of Administrator Passwords

The TOE stores administrative password data in a salted SHA-256 hash within an internal configuration file. Further, the TOE does not provide any way for an administrator to view, extract, or read the password. The TOE only accepts passwords during authentication attempts (or during administrative changing of the password), salts and hashes the provided password (and then either compares it to the stored value or stores the new value depending upon whether the administrator is authenticating or changing the administrative password).

This aspect of the TSF Protection security function satisfies FPT_APW_EXT.1.

5.5.2 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

The TOE persistently stores a private key that is used for its TLS server certificate. The TOE stores this key internally and does not provide any command to access this key, and the TOE only accesses the key for use with TLS. The key is stored in an encrypted format but the specific encryption method was not evaluated as the SFR does not require encrypted key storage; only that no interface exists to deliberately

provide access to stored key data. The key storage is assumed to provide equivalent strength to cleartext. As part of establishing TLS sessions, the TOE also generates and exchanges a pre-master secret and session key for each session, which is maintained in memory for as long as they are needed to perform their associated operations. These keys are stored in memory as cleartext and there is no direct interface by which they can be accessed.

This aspect of the TSF Protection security function satisfies FPT_SKP_EXT.1.

5.5.3 TSF Testing

The TOE performs a series of power-up Known Answer Tests (a KAT for each library cryptographic algorithm that the TOE utilizes) as well as an integrity test for its cryptographic module during power-up (enumerated in section 4.2.5.4). For each self-test, the TOE uses known inputs to calculate an expected cryptographic result, and compares that result to the known result. If the calculated result matches the expected result, the test passes; if it does not match, the test fails. A failure of a power-up self-test causes the TOE to halt its boot.

The TOE also uses AES-XTS to protect TSF data at rest. Upon power on, the stored data is decrypted. Since there is no way to gain general-purpose access to the TOE while it is in an operational state, there is no mechanism by which decrypted data at rest can be illicitly modified. Any attempted modification of encrypted data at rest would cause the decrypted data to be corrupted in such a manner that the TOE would not boot into an operational state.

These self-tests are collectively sufficient to ensure that when the TOE is presenting itself to an administrator as being in an operational state, it is not operating in an insecure manner without the administrator's awareness.

This aspect of the TSF Protection security function satisfies FPT_TST_EXT.1.

5.5.4 Trusted Update

The TOE provides an administrator the ability to view the current version of firmware (for example, on the Administration tab of the Administration Tool, one can see the 'Firmware: ' version) as well as to request that the TOE update its firmware.

As a prerequisite, the administrator first needs to obtain the new TOE firmware (in the form of a *.bau package that contains .rpm files) and then, using the Administration Tool, host that firmware on the workstation (using the Update Server tab), in order for the TOE to perform the update.

The TOE will download the new firmware, verify the digital signature on each update component, and only install the updates if the signatures verify. A 3072-bit RSA digital signature is used to ensure the integrity of all RPMs.

This aspect of the TSF Protection security function satisfies FPT_TUD_EXT.1.

5.5.5 Reliable Time Stamps

The TOE utilizes time when creating audit records and when checking syslog server and administrative client certificates (for expiration and revocation), for session inactivity timeout, and for administrator lock out periods. The TOE obtains and maintains time by allowing the administrator to both manually specify the time as well as configure an NTP server with which the TOE synchronizes its clock. Specifically, the manually-configured time is specified through a synchronization option that allows the TOE's clock to be

set to the same time as the system clock of the administrative workstation being used to access the TOE. The TOE also has a battery-backed real-time clock that is used to guarantee the availability of time data.

This aspect of the TSF Protection security function satisfies FPT_STM_EXT.1.

5.6 TOE Access

5.6.1 Access Banner

The TOE provides the administrator the ability to set a banner that the TOE displays before each local and remote administrator login. Local and remote administrator logins occur through the TOE's TLS-protected management port, and the TOE will send the banner to the administrator in the same fashion (returning the configured banner through the established TLS connection to the administrator's client application).

This aspect of the TOE Access security function satisfies FTA_TAB.1.

5.6.2 Session Termination

The administrator can configure a value between 60 and 600 (with the default being 60) for the timeout of the administrative session in seconds. The configuration will take effect during the next administrative session. The timeout applies to both local and remote sessions, which are terminated by the TOE when the inactive timeout value is reached.

The Administrator can connect and disconnect their administrative session from within the remote application from which the interactive session was established. Likewise, the Administrator can manually terminate their own local administrator session.

This aspect of the TOE Access security function satisfies FTA_SSL_EXT.1, FTA_SSL.3, and FTA_SSL.4.

5.7 Trusted Path/Channels

The TOE provides the administrator the ability to export audit records to a TLS-protected syslog server. The TOE's TLS syslog configuration allows the administrator to specify the root CA certificate (as well as any needed intermediate CA certificates and any CRLs for revocation) for the TOE to use when validating the syslog server's certificate.

The administrator can connect to the TOE using its TLS/HTTPS protected administration channel.

The Trusted Path/Channels security function satisfies FTP_ITC.1 and FTP_TRP.1/Admin.

6 Protection Profile Claims

This ST conforms to the collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 [cPPND] and including the following selection-based SFRs: FCS_NTP_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; and FMT_MOF.1/Functions.

As explained in Section 2, Security Problem Definition, the Security Problem Definition of the [cPPND] has been included by reference into this ST.

As explained in Section 3, Security Objectives, the ST reproduces the security objectives for the operational environment from [cPPND].

As explained in Section 4, IT Security Requirements, the SFRs have all been drawn from [cPPND]. As such, operations on SFRs already performed in that PP are not identified in this ST. Rather, the SFRs have been copied from [cPPND] and any formatting used in that PP has been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs for the TOE are included by reference from the [cPPND].

7 Rationale

This ST includes by reference the [cPPND] Security Problem Definition and SARs and reproduces the security objectives for the Operational Environment. The ST makes no additions to the [cPPND] assumptions. [cPPND] SFRs have been reproduced with the Protection Profile operations completed. Operations on the SFRs follow [cPPND] application notes and assurance activities. Consequently, [cPPND] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

7.1 TOE Summary Specification Rationale

Each subsection in Section 5, the TOE Summary Specification (TSS), describes a security function of the TOE. Each description identifies the SFRs that are covered by that description and, as such, provides the rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section, in conjunction with the TSS, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TSS are all necessary for the required security functionality in the TSF. Table 9: Security Functions vs. Requirements Mapping summarizes the relationship between security requirements and security functions.

Table 9: Security Functions vs. Requirements Mapping

Specification	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG_EXT.1	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1/DataEncryption		X					
FCS_COP.1/SigGen		X					
FCS_COP.1/Hash		X					
FCS_COP.1/KeyedHash		X					
FCS_NTP_EXT.1		X					
FCS_RBG_EXT.1		X					

Specification	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FCS_TLSC_EXT.1		X					
FCS_TLSS_EXT.1		X					
FIA_AFL_EXT.1			X				
FIA_PMG_EXT.1			X				
FIA_UIA_EXT.1			X				
FIA_UAU_EXT.2			X				
FIA_UAU.7			X				
FIA_X509_EXT.1/Rev			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FMT_MOF.1/Functions				X			
FMT_MOF.1/ManualUpdate				X			
FMT_MTD.1/CoreData				X			
FMT_SMF.1				X			
FMT_SMR.2				X			
FPT_APW_EXT.1					X		
FPT_SKP_EXT.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FPT_STM_EXT.1					X		
FTA_SSL_EXT.1						X	
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1/Admin							X