One Identity Safeguard for Privileged Passwords v6.7

Security Target

Version 1.0

2021-4-12

Prepared for:

4 Polaris Way
Aliso Viejo, CA 92656
United States

Prepared by:

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

Revision History		
Date	**Author**	**Modifications**
2021-4-12	Leidos	Updates based on evaluation findings
2021-2-11	Leidos	Updates based on vender input and ECRs
2020-11-02	Leidos	Updates based on AAR activity
2020-08-10	Leidos	Updates based on vendor input
2020-07-27	Leidos	Updates based on ASE activity
2020-06-29	Leidos	Updates based on initial review
2020-06-17	Leidos	Initial draft

Table of Contents

List of 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: One Identity Safeguard for Privileged Passwords v.6.7 Security Target

ST Version: Version 1.0

ST Date: 2021-4-12

TOE Identification: One Identity Safeguard for Privileged Passwords v6.7, comprising the following hardware and firmware:

- Safeguard for Privileged Passwords 3000 Appliance
- Safeguard for Privileged Passwords firmware v6.7

TOE Developer: One Identity LLC

Evaluation Sponsor: One Identity LLC

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: FCS_HTTPS_EXT.1/Client, FCS_HTTPS_EXT.1/Server, FCS_TLSC_EXT.1; FCS_TLSC_EXT.2; FCS_TLSS_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MTD.1/CryptoKeys and FMT_MOF.1/Functions.

The following NIT Technical Decisions and NIAP Interpretations apply to this [cPPND] and have been accounted for in the ST development and the conduct of the evaluation:

- TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)
- TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4
- TD0536: NIT Technical Decision for Update Verification Inconsistency
- TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3
- TD0538: NIT Technical Decision for Outdated link to allowed-with list
- TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63
- TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN
- TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test
- TD0556: NIT Technical Decision for RFC 5077 question
- TD0563: NiT Technical Decision for Clarification of audit date information
- TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria
- TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7
- TD0570: NiT Technical Decision for Clarification about FIA_AFL.1
- TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1
- TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers
- TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e
- TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3

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.
- Operations completed in the PP are not reproduced in this ST (i.e. they are unformatted).

1.4 Abbreviations and Acronyms

Table 1: Abbreviations and Acronyms

Acronym	Definition
ACF2	Access Control Facility
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DH	Diffie-Hellman
DPAPI	Data Protection API
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MAC	Message Authentication Code
PEM	Privacy-Enhanced Mail
PBKDF2	Password-Based Key Derivation Function 2
REST	Representational State Transfer
SHA	Secure Hash Algorithm
SPP	Safeguard for Privileged Passwords
SPS	Safeguard for Privileged Sessions- a privileged session management solution, which provides access control, session recording and auditing to prevent privileged account misuse and accelerate forensics investigations.
SSH	Secure Shell
TPM	Trusted Platform Module
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
UI	User Interface

1.5 TOE Overview

The TOE is One Identity Safeguard for Privileged Passwords v6.7, comprising a standalone hardware appliance: One Identity Safeguard for Privileged Passwords 3000 Appliance with Safeguard for Privileged Passwords firmware v6.7. One Identity Safeguard for Privileged Passwords automates, controls, and secures the process of granting privileged credentials with role-based access management and automated workflows.

1.6 TOE Description

One Identity Safeguard for Privileged Passwords (Safeguard or SPP) is pre-installed firmware on a hardened network appliance and is used to automate the issuance and management of privileged passwords for various organizational assets. Hardening is accomplished through use of a Trusted Platform Module (TPM) to ensure the integrity of the TOE firmware and secure-by-default configuration that prevents any users or IT entities from interacting with the TOE's OS platform as a general-purpose computer.

The product communicates out to any number of managed systems (assets) of different types and ensures that password data is up-to-date based on policies provisioned internally on the product. An asset is a computer, server, network device, or application managed by a Safeguard for Privileged Passwords Appliance. Supported asset types include ACF2, Active Directory, AIX, Amazon Linux, AWS, CentOS, etc… The TOE supports secure communication channels with many of these assets via HTTPS, or TLS. The solution also supports secure communication with assets using SSH. However, when the TOE administrator executes an operation on SPP that requires an SSH connection to be opened up, the SPP Desktop Client (non-TOE) launches the SSH client on the administrator's local machine. Therefore the TOE is not initiating an SSH client connection but rather this is being done in the operational environment. Section 8 lists the asset types and versions with which the TOE can communicate over HTTPS or TLS, the protocols covered by the evaluation. The evaluation did not cover the functionality related to the issuance and management of privileged passwords, other than the functions for securing the transmission channels.

For the purpose of this evaluation, the TOE is evaluated as a network device offering CAVP certified cryptographic functions, security auditing, secure administration, trusted updates, self-tests, and secure connections with other servers (i.e. assets, and syslog), protected using TLS or HTTPS. The evaluation is restricted to the functions and claims made in conformance with the cPPND.

1.7 Physical Scope

The TOE is One Identity Safeguard for Privileged Passwords v6.7, comprising the Safeguard for Privileged Passwords 3000 Appliance with Safeguard for Privileged Passwords firmware v6.7. The TOE is evaluated as a single standalone network device and cluster configurations are not included.

The TOE includes a Windows 10 IoT Enterprise operating system with Bitlocker v2.0, a TPM (spec v2.0) -- IFX 5.62.3126.0, and an Intel Xeon E3-1275 v6 processor with Kaby Lake microarchitecture.

The TOE provides Web UI and REST API management interfaces that an administrator can access via Ethernet ports. The desktop client application uses the TOE's REST API for TOE administration. The desktop client software is provided with the TOE and is considered part of the operational environment.

Local management of the TOE is possible by directly connecting the administrator's laptop to the appliance's XO port via Ethernet cable. The management interfaces are protected with HTTP over TLS (HTTPS) and are limited to the authorized administrator as defined by their designated assigned role(s). The computer hosting the desktop client application is part of the operational environment and required to have the desktop client software. The administrator can physically connect to the serial port to use the Recovery Kiosk, which takes it out of its evaluated configuration. The Recovery Kiosk can be used to enable the Bootstrap Administrator account in order to then unlock a locked admin account if all accounts have been locked (e.g., due to consecutive failed authentication attempts locking the admin accounts). After the Bootstrap Administrator account has been used to unlock an admin account, the Bootstrap Administrator account is disabled, which returns the TOE to its evaluated configuration. Regardless of the method used to access the TOE appliance, no general-purpose computing interface (e.g. Remote Desktop, PowerShell) is available.

The TOE provides a local password-based and X.509 certificate based identification and authentication methods. No other authentication methods offered by the product (e.g., LDAP, AD, external authentication servers) were covered in the evaluation and their use is excluded in the evaluated configuration.

Depending on configuration, the TOE in its evaluated configuration may require the following components in its operational environment:

- A TLS-protected syslog server that receives audit events from the TOE,
- Supported organizational assets (ACF2, Active Directory, AWS, Windows, etc.),
- Desktop Client workstation for administrator access to API.
    - A supported operating system: Windows 2008 Server, Windows 7, Windows 2012 Server, Windows 2012 R2 Server, Windows 8, Windows 8.1, Windows 10, Windows 2016, or any recent version of Linux.
    - SPP Desktop Client software version 6.7, and
    - SSH Client software.
- A supported browser for access to the web UI: Mozilla Firefox (minimum version 69); Google Chrome (minimum version 80); or Microsoft Edge (Chromium-based, minimum version 80). The browser must support TLS-encrypted HTTPS connections, and JavaScript/cookies must be enabled. For any of the supported browsers, the latest patched version should be used.

The following features and capabilities are not covered by the evaluation:

- The product's functionality for automating, controlling, and securing the process of granting privileged credentials with role-based access management and automated workflows.
- The use of the product to automate the issuance and management of privileged passwords for organizational assets.
- The product's ability to ensure that password data is up-to-date based on policies provisioned internally on the product was not covered by the evaluation.

The table below identifies additional features or protocols that are not evaluated or must be disabled and the rationale why.

Feature	Description
SPP Serial Port	SPP has a serial port used for Recovery Kiosk functionality. The interface itself is unauthenticated but some functions are challenge-response to One Identity (e.g. reset admin password). This is not included in the evaluated configuration.
Recovery Kiosk	The use of Recovery Kiosk is permitted only to enable the built-in Bootstrap Administrator account in order to then unlock a locked admin account in the event all admin accounts are locked.
Telnet and Custom HTTP	Telnet and HTTP are disabled by default and must not be enabled in the evaluated configuration. Telnet and HTTP are insecure protocols, which allow for plaintext passwords to be transmitted.
Archive Server	The use of an archive server for storing backup files, session recordings, and audit records is out of the scope of the evaluation.
External Authentication Servers	Starling, Active Directory, LDAP, RADIUS, and FIDO2 are excluded from the evaluation and should not be used.
Communication with SPS	SPP is capable of providing credentials to Safeguard for Privileged Sessions (SPS) in the operational environment for secondary non-admin user authentication. This capability was not evaluated and in the evaluated configuration this interface must be disabled.
Communication with Assets using SSH	SSH is supported by the solution but was not evaluated and is outside the TOE boundary.
NTP	Although the product supports NTP, the use of NTP was not evaluated and it is excluded from the evaluated configuration.
Safeguard for Privileged Passwords virtual appliances and cloud applications	One Identity Safeguard for Privileged Passwords may be deployed as virtual appliances and cloud applications. However, neither of these configurations were tested and are not in the evaluated configuration.

1.8 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.

1.8.1 Security Audit

The TOE generates security relevant audit records, stores them locally, and can be configured to forward them to a syslog server over TLS. The locally stored audit records are protected from unauthorized access.

1.8.2 Cryptographic Support

The TOE includes FIPS-approved cryptographic libraries with CAVP certificates for their cryptographic algorithms. The TOE uses its Windows cryptographic libraries for all HTTPS, TLS and certificate functionality. Cryptographic services include key management, random bit generation, symmetric encryption and decryption, digital signature, and secure hashing.

1.8.3 Identification and Authentication

The TOE displays a configurable warning banner, and allows automated generation of cryptographic keys prior to the user being successfully identified and authenticated. No other actions are permitted until the user is authenticated. The TOE provides username/password and X.509 certificate based identification and authentication methods, password management functions, and authentication failure management functions.

1.8.4 Security Management

The TOE provides Web UI and REST API management interfaces that an administrator can access via a network port. The TOE's REST API can be accessed from the desktop client application or may be invoked directly if desired. Local management of the TOE is possible by directly connecting the administrator's laptop to the appliance's XO port via Ethernet cable. The management interfaces are protected with HTTP over TLS (HTTPS) and are limited to the authorized administrator as defined by designated assigned role(s).

1.8.5 Protection of the TSF

The TOE implements features designed to protect itself to ensure the reliability and integrity of its security features to include protecting sensitive data; and providing timing mechanisms to ensure that reliable time information is available (e.g., for log accountability).

The TOE includes functions to perform self-tests so that it can detect when it is failing and transition to a secure, quarantine state. It also includes a mechanism to verify TOE updates to prevent malicious or other unexpected changes in the TOE.

1.8.6 TOE Access

The TOE displays a Security Administrator-specified advisory notice and consent warning message prior to establishing an administrative user session. The TOE terminates local and remote administrator interactive sessions after a Security Administrator-specified time period of inactivity. The TOE allows administrator-initiated termination of the administrator's own interactive session.

1.8.7 Trusted Path/Channels

The TOE provides trusted paths and channels for remote administrators and trusted IT entities. The TOE can be configured to send audit records to external syslog server(s) using TLS in real-time.

1.9 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:

- One Identity Safeguard for Privileged Passwords 6.7 Common Criteria Evaluated Configuration Guide (CCECG), Version 1.0, April 12, 2021
- One Identity Safeguard for Privileged Passwords 6.7 Administrator Guide, Updated 28 August 2020
- One Identity Safeguard for Privileged Passwords 6.7 User Guide, Updated 28 August 2020
- One Identity Safeguard for Privileged Passwords 6.7 Appliance Setup Guide, Updated 28 August 2020

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.

3 Security Objectives

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

3.1 Security Objectives for the Operational Environment

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.

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
- FCS_HTTPS_EXT.1—HTTPS 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
- 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/Rev—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_EXT.1—Protected audit event storage

– Requirement Class	– Requirement Component
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/Client—HTTPS Protocol (Client)
	FCS_HTTPS_EXT.1/Server —HTTPS Protocol (Server)
	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
FIA: Identification and authentication	FIA_AFL.1—Authentication failure management
	FIA_PMG_EXT.1—Password management
	FIA_UIA_EXT.1—User identification and authentication
	FIA_UAU_EXT.2—Password-based authentication mechanism
	FIA_UAU.7—Protected authentication feedback
	FIA_X509_EXT.1/Rev—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: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1/Client	Failure to establish a HTTPS Session	Reason for failure
FCS_HTTPS_EXT.1/Server	Failure to establish a HTTPS Session	Reason for failure
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
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	Reason for failure of certificate validation.
	Any addition, replacement or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store.
FIA_X509_EXT.2	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	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).
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
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.	None.

Requirement	Auditable Events	Additional Audit Record Contents
	Failure of the trusted path functions	

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 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 [***[cease to operate]***] when the local storage space for audit data is full.

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] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;***

].

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: [

- ***Elliptic curve-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 [***destruction of reference to the key directly followed by a request for garbage collection***];

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

that meets the following: No Standard.

Application Note: There are no plaintext keys in non-volatile storage and therefore that portion of the requirement is vacuously met.

4.2.2.4 Cryptographic Operation (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 (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, 4096 bits],***

- ***Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 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]; ISO/IEC 14888-3, Section 6.4***

].

4.2.2.6 Cryptographic Operation (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 (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, 1024 bits]** and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

4.2.2.8 HTTPS Protocol (Client) (FCS_HTTPS_EXT.1/Client)

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

FCS_HTTPS_EXT.1.2/Client The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3/Client If a peer certificate is presented, the TSF shall [***[enforce administrator configuration to either establish the connection or to not establish the connection]***] if the peer certificate is deemed invalid.

Application Note: *The HTTPS Client protocol is used for connections with the Assets.*

4.2.2.9 HTTPS Protocol (Server) (FCS_HTTPS_EXT.1/Server)

FCS_HTTPS_EXT.1.1/Server The TSF shall implement the HTTPS protocol **when acting as a web server** that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Server The TSF shall implement HTTPS using TLS.

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

Application Note: *The HTTPS protocol is used when SPP is acting as a web server (UI and API).*

4.2.2.10 Random Bit Generation (FCS_RGB_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] software-based noise source,*]** with a minimum of **[*256 bits*]** of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

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)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***

] and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 section 6] and no other attribute types***].

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 [

- ***require administrator authorization to establish the connection if the TSF fails to [match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate***

].

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

Application Note: *The TLS Client protocol is used for outbound connections to assets and the audit server.*

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.

Application Note: *The TLS Client protocol with mutual authentication is optionally supported for syslog server.*

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)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

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

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***TLS 1.1***].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [***ECDHE curves [secp256r1] and no other curves***].

FCS_TLSS_EXT.1.4 The TSF shall support [***session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)***].

Application Note: *The TLS server protocol without mutual authentication is used for the web UI/REST API.*

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 [**1-100**] 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 remote Administrator from successfully establishing a remote session using any authentication method that involves a password until [manual unlock] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

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: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [ASCII decimal 32 to 126]***];
b) Minimum password length shall be configurable to between [**3**] and [**255**] characters.

4.2.3.3 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;
- [***automated generation of cryptographic keys***].

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.

4.2.3.4 Password-Based Authentication Mechanism (FIA_UAU_EXT.2)

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

4.2.3.5 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.

4.2.3.6 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 certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:

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

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

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 [***not accept the certificate***].

4.2.3.8 X.509 Certificate Requests (FIA_X509_EXT.3)

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

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/Functions)

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

4.2.4.2 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.

4.2.4.3 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 start and stop services;***
- ***Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);***
- ***Ability to modify the behaviour 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 re-enable an Administrator account;***
- ***Ability to set the time which is used for time-stamps;***
- ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;***
- ***Ability to import X.509v3 certificates to the TOE's trust store;***

].

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 symmetric and private keys) (FPT_SKP_EXT.1)

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

4.2.5.3 Reliable Time Stamps (FPT_STM.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***].

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: **[Cryptographic Primitive tests, and Firmware Integrity tests)**].

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, HTTPS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[assets]***] 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 [**auditing, password management**].

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 administrative actions.

4.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the [CPP_ND_V2.2E].

Table 5: Assurance Components

Requirement Class	Requirement Component
ASE: Security Target	ASE_CCL.1: Conformance claims
	ASE_ECD.1: Extended components definition
	ASE_INT.1: ST introduction
	ASE_OBJ.1: Security objectives for the operational environment
	ASE_REQ.1: Stated security requirements
	ASE_SPD.1: Security Problem Definition
	ASE_TSS.1: TOE summary specification
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 chapter describes the following security functions:

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

5.1 Security Audit

SPP generates security relevant audit records, stores them locally, and can be configured to forward them to a syslog server over TLS. The locally stored audit records are protected from unauthorized access.

5.1.1 FAU_GEN.1: Audit Data Generation

The TOE generates log records for security relevant events as they occur.

The events that can cause an audit record to be logged include starting and stopping the audit function (in parallel with the device itself so device startup/shutdown is logged for this) [1], administrator login/logout, changes to TSF data related to configuration changes (Web Interface, or REST API), management of key data, and password changes as well as all of the events identified in Table 44.

Management of key data causing an audit record to be logged consists of: creating a new server certificate; adding/removing trusted certificate, Server certificate and SSL certificates; and creating/deleting server signature request.

All log records include the following contents: date/time; event type; user ID (i.e., username, IP address) or component (i.e., syslog); and description of the event including success or failure. For user-initiated actions, the User ID is included in the log records. For cryptographic key operations, the key name or reference is also logged. When creating new server certificates, the Certificate Signing Request (CSR) is logged. Additionally and based on the event, the description of the event will include additional information as required in Table 4.

5.1.2 FAU_GEN.2: User Identity Association

The TOE identifies the responsible user for each event based on the specific username and/or network entity (identified by source IP address) that caused the event.

[1] The audit function is always on when the TOE is operational. There is no command to start/stop the audit function apart from starting/stopping the TOE.

5.1.3 FAU_STG_EXT.1: Protected Audit Event

The Appliance admin can configure an audit retention period, which causes oldest logs to be deleted. The TOE does not provide any interfaces to modify or delete individual audit logs. The TOE restricts access to the audit retention function using role-based access control. The retention period is not enabled by default and when enabling the function, the number of days after which the audit records are purged must be specified.

The TOE does not have separate storage for audit data. When the TOE's audit space is exhausted, the overall disk space is also exhausted. This would prevent normal functioning of the product and the TSF would cease to operate. However, the amount of disk space available on the TOE means this is unlikely to occur. The Safeguard 3000 appliance has one 3.43 TB logical disk. Based on storage requirements of various system configurations (e.g. TOE firmware, underlying OS, components, configuration and debug data), the estimated amount of disk available for local audit data ranges between 1 GB minimum and 3.30 TB maximum. Though it is unlikely that the audit space would be exhausted, this can be further reduced by setting the retention period to a small number of days rather than a large number of days.

The TOE is a single standalone component that stores audit data locally and can be configured to transmit the generated audit data to an external syslog server using TLS. Data is written to the external syslog in real time.

5.2 Cryptographic Support

The TOE includes Microsoft Windows 10 IoT Enterprise FIPS-approved cryptographic libraries (i.e. cng.sys, bcryptprimitives.dll, ncryptsslp.dll). Each of these libraries possesses CAVP certificates for their different cryptographic algorithms. Table 6 below summarizes the CAVP certificates.

The TOE uses its Windows cryptographic libraries for all HTTPS, TLS and certificate functionality.

Table 6: Cryptographic Functions

Functions	Standards	Certificates
Asymmetric key generation (FCS_CKM.1)		
RSA Schemes (2048-bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	RSA #C2150
ECC Schemes (ECDSA P-256, P-384 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	ECDSA #C2150
Key Establishment (FCS_CKM.2)		
Elliptic curve-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography	KAS-ECC #C2150
Encryption/Decryption (FCS_COP.1/Data Encryption)		
AES in CBC mode (128, 256 bits)	CBC as specified in ISO 10116	AES #C2150
AES in GCM mode (128, 256 bits)	GCM as specified in ISO 19772	AES #C2150
Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)		

Functions	Standards	Certificates
RSA Digital Signature Algorithm (rDSA) (2048/4096 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,	RSA #C2150
ECDSA Elliptic Curve Digital Signature Algorithm (P-256, P-384 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-256, P-384; ISO/IEC 14888-3, Section 6.4	ECDSA #C2150
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 #C2150
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) HMAC-SHA-512 (key size 1024 bits, digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	HMAC #C2150
Random bit generation (FCS_RBG_EXT.1)		
CTR-DRBG (AES) – minimum 256 bits entropy	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions	DRBG #C2150

5.2.1 FCS_CKM.1: Cryptographic Key Generation (for Asymmetric Keys)

The TOE generates asymmetric keys for TLS and X.509 certificates.

RSA schemes using cryptographic key sizes of 2048 bits that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 are for use with TLS and X.509 certificates.

ECC schemes using P-256 and P-384 NIST curves that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 are for use with TLS.

5.2.2 FCS_CKM.2: Cryptographic Key Establishment (Refined)

The TOE performs cryptographic key establishment in accordance with the following specified cryptographic key establishment method.

Elliptic curve-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".

The TOE uses the key establishment scheme as identified in Table 6 for the services identified in the table below.

Table 7: Key Establishment Schemes and Usage

Scheme	SFR	Service
Elliptic curve-based	FCS_TLSC_EXT.1 FCS_TLSS_EXT.1	Assets, Audit Server Administration

5.2.3 FCS_CKM.4: Cryptographic Key Destruction

For plaintext keys in volatile storage, the destruction of keys is performed by the Windows OS and executed by a destruction of reference to the key, directly followed by a request for garbage collection using the **`RtlSecureZeroMemory`** function provided by Windows. When ephemeral keys or secrets are no longer needed (e.g. a network session has terminated), they are deleted.

There are no plaintext keys in non-volatile storage and therefore that portion of the requirement is vacuously met. The Windows Certificate store is used for TLS certificate and private key storage. All other permanently stored keys/secrets are stored in the Windows file system. All keys and CSPs are stored encrypted by DPAPI using 256-bit AES in CBC mode, SHA512 for hashing and PBKDF2 as the password-based key derivation routine. The DPAPI key is the Key Encrypting Key used for encrypting/decrypting the keys. Factory resetting the device erases user accounts, which wipes the DPAPI keys for those accounts. The TOE derives DPAPI keys from user logon secrets and administrator credentials are provided by the user. All other keys and CSPs originate from the cryptographic libraries.

Table 8: Secret keys, Private keys and CSPs

Key/CSP	Storage Location	Zeroized upon:	Zeroized by:
RSA private key (TLS)	In memory (volatile RAM)	Handshake done	Crypto library
ECDSA private key (TLS)	In memory (volatile RAM)	Handshake done	Crypto library
DRBG entropy input/seed/key	In memory (volatile RAM)	When a new random value is needed	Crypto library
Diffie-Hellman private key and shared secret	In memory (volatile RAM)	No longer needed	Crypto library

Key/CSP	Storage Location	Zeroized upon:	Zeroized by:
ECDH private key and shared secret	In memory (volatile RAM)	No longer needed	Crypto library
Administrator credentials	File system	Zeroized by overwriting with new password	Operating System
TLS pre-master secret, session key, and session authentication key	In memory (volatile RAM)	Close of session	Crypto library
TLS private key	Windows certificates store	Command: CryptAcquireContext with the CRYPT_DELETE_KEYSET flag	Operating System
DPAPI Key - Credential for X.509 trust store (for TLS)	File System	Factory reset	Operating System

5.2.4 FCS_COP.1/1: Cryptographic Operation (Data Encryption/Decryption)

The TOE supports AES CBC, and AES GCM (128 and 256 bits) for data encryption/decryption. The algorithms are implemented according to the following standards: AES as specified in ISO 18033-3, CBC and GCM as specified in ISO 19772.

Both AES CBC and AES GCM use both key sizes and are used for TLS.

5.2.5 FCS_COP.1/2: Cryptographic Operation (Signature Generation and Key Verification)

The TOE supports rDSA (modulus 2048/4096) and ECDSA with elliptical curve key sizes 256 or 384 bits for signature generation and verification. The algorithms meet the following standards:

- 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 P-256, P-384 “NIST curves”; ISO/IEC 14888-3, Section 6.4.

5.2.6 FCS_COP.1/3: Cryptographic Operation (Hash Algorithm)

The TOE performs SHA-1, SHA-256, SHA-384, and SHA-512 cryptographic hashing services that meets ISO/IEC 10118-3:2004.

The SHA-1/256/384 hash functions are used with the following cryptographic functions: FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, and as part of HMAC and RSA digital signature creation and verification.

The SHA-512 hash functions is used for securely storing keys and CSPs in non-volatile memory.

5.2.7 FCS_COP.1/4: Cryptographic Operation (Keyed Hash Algorithm)

The TOE performs keyed-hash message authentication that meets the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2. The hash function used, key and block size, and output MAC lengths (message digest size) are identified in the table below.

Table 9: Keyed Hash MAC and Sizes

Algorithm	Key Size	Block Size	Message Digest Size
SHA-1	160	512	160
SHA-256	256	512	256
SHA-384	384	1024	384
SHA-512	512	1024	512

5.2.8 FCS_HTTPS_EXT.1/Client: HTTPS Protocol (Client), FCS_HTTPS_EXT.1/Server HTTPS Protocol (Server)

The TOE implements HTTPS for web-based secure administrator sessions (UI and API) and connections with assets. The HTTPS protocol complies with RFC 2818 and is implemented using TLS.

When acting as a web server, if mutual authentication is used, the product will always reject an invalid client certificate presented to it and not establish the connection.

For any outbound HTTPS connections to assets, each connection (i.e. per-host) is individually configurable as to whether an invalid certificate should be accepted or rejected. This is done at a global level (e.g. you can't say "allow expired certificate but deny revoked certificate"). Configuration of this is done by Asset Admin. When an invalid peer certificate is presented, the TOE enforces the administrator's configuration to either establish the connection or to not establish the connection.

5.2.9 FCS_RBG_EXT.1 Random Bit Generation

The TOE includes Microsoft Windows 10 IoT Enterprise OS, which provides the TOE's cryptographic capabilities. Windows 10 includes the following cryptographic libraries:

- Kernel Mode Cryptographic Primitives Library (cng.sys)—runs as a kernel mode export driver that provides cryptographic services to Windows 10 kernel components
- Cryptographic Primitives Library—comprises two Dynamic Link Library (DLL) files (bcryptprimitives.dll, ncryptsslp.dll) and provides cryptographic services to components and applications running on Windows 10. As such, the TOE relies on entropy that is collected and conditioned using the cryptographic capabilities provided by Windows 10, specifically through bcryptprimitives.dll and cng.sys.

The vendor-developed SPP firmware is implemented on the .NET framework and uses the Cryptography API: Next Generation (CNG) to access the cryptographic services of the Cryptographic Primitives Library.

The ISO/IEC 18031:2011 Deterministic Random Bit Generator (DRBG) used by the TOE is AES-256 counter mode (CTR_DRBG (AES)). It is implemented in bcryptprimitives.dll and invoked from the TOE application via the BCryptGenRandom() function. Invocation of BCryptGenRandom() instantiates the DRBG with a 256-bit seed obtained via the SystemPrng() function from the in-kernel random number generator implemented in cng.sys (i.e. one software-based source). The software-based entropy source is described in the proprietary Entropy Documentation. The TOE uses this DRBG to generate all keys as well as to generate salts for password hashing. The entropy source is assumed to provide a full 256 bits of entropy at minimum.

5.2.10 FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication

The TOE implements TLS 1.2 (RFC 5246) for TLS Client communications and rejects all other TLS and SSL versions. In the evaluated configuration, the administrator must configure the TOE to use TLS 1.2. When this configuration is done, both 1.1 and 1.0 are disabled.

The TLS client protocol is used with assets and the audit server (mutual authentication is optional) and supports the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The TOE's TLS client implementation establishes its reference identifiers from the administrator-configured reference identifiers per Section 6 of RFC 6125, using the DNS in the CN or SAN and checking that the server's certificate includes the specified identifier. The TOE supports handling of wildcards within certificates. The TLS client supports the Elliptic Curves Extension (specifying only P-256 or P-384) in its Client Hello, and the TOE does not require (nor allow) administrative configuration of this extension. Certificate pinning is not used by the TLS client.

When establishing a trusted channel, by default the TOE will not establish a trusted channel if the server certificate is invalid. The Administrator has the option to disable rejection for outbound communications with assets only. This is toggled globally on a per-connection basis, not on specific validation failure types. i.e. it is possible to make it so that one connection enforces certificate rejection and another does not, but it is not possible to define what causes certificate rejection on a granular level (i.e. failure to match the reference identifier, failure to validate certificate path, failure to validate expiration date, or failure to determine the revocation status).

The TOE presents the Supported Elliptic Curves Extension with the secp256r1 and secp384r1 NIST curves in the Client Hello.

5.2.11 FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication

The TOE supports TLS communication with mutual authentication using X.509v3 certificates for communications with the audit server. Mutual authentication is optional for this connection and when configured for mutual authentication does not permit the configuration of the certificate failure override feature.

5.2.12 FCS_TLSS_EXT.1: TLS Server Protocol Without Mutual Authentication

The TOE implements TLS 1.2 (RFC 5246) for TLS Server communications and rejects all other TLS and SSL versions. In the evaluated configuration, the administrator must configure the TOE to use TLS 1.2. When this configuration is done, both 1.1 and 1.0 are disabled.

The TLS server protocol without mutual authentication is used for the web UI/REST API. The supported cipher suites are not individually configurable and are specified below.

- 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 performs key establishment for secp256r1 ECDHE curves and supports session resumption based on session IDs according to RFC 5246 (TLS1.2). The TOE performs session resumption when it receives a Session ID in a Client Hello that is the same as the session ID used in a previous session.

5.3 Identification and Authentication

The TOE displays a configurable warning banner and allows automatic key generation and no other actions until the user is identified and authenticated. The TOE provides username/password and X.509 certificate based identification and authentication methods, password management functions, and authentication failure management functions.

5.3.1 FIA_AFL.1: Authentication Failure Management

The TSF implements a counter mechanism to keep track of and detect when an Administrator configurable positive integer within 1-100 (default 5) unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password. If a user submits an incorrect password for the maximum number of times specified by the account **Lockout Threshold** settings within the **Lockout Window**, the TOE locks the account until the configurable **Lockout Duration** period of between 1 and 9,999 minutes (default 15 minutes) has been met; or until the account has been manually unlocked by an administrator. The settings are applied to both the Web UI and API HTTPS administration connections.

In the event all active administrators are locked out due to the consecutive failed authentication attempts mechanism, an administrator can unlock locked out users by connecting to the appliance's serial port and using the Recovery Kiosk to re-activate the built-in Bootstrap Administrator account. The administrator can then login to the TOE and unlock the admin accounts. Administrator guidance directs the administrator to again disable the Bootstrap Administrator account after it has been used to unlock an admin account.

5.3.2 FIA_PMG_EXT.1 Password Management

The TOE provides password management rules that govern the construction of passwords. Rules can be created to control the allowable password length (from three to 225 characters); and character types allowed (any combination of upper and lower case letters, numbers, and special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ASCII decimal 32 to 126). Password length and number of/type of characters are configurable. Admin can whitelist and blacklist special characters. Out of the box, the default password character range is 8 to 64, but both floor and ceiling are configurable.

5.3.3 FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-Based Authentication Mechanism. FIA_UAU.7: Protected Authentication Feedback

The administrative functions are accessed via a Web UI and an API, locally or remotely both using HTTP over TLS (HTTPS).

Only the following actions are permitted prior to the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- Automated generation of cryptographic keys (for establishment of TLS sessions).

The TOE requires each administrative user to be successfully identified and authenticated using the TOE's local password-based authentication mechanism before allowing any other TSF-mediated actions on behalf of that administrative user. A user logon is successful when the username and password entered by the user matches the credentials stored in the TOE's database. For the API, the username and password are provided in the form of a token. Specifically, the administrator requests a secure token service (STS) token by issuing an HTTPS request that includes the relevant username/password data. Once validated by the TSF, the TOE will issue an API token, which is a JSON Web Token bearer token. This token is subsequently used as authorization to execute the requested functions over the REST API. Password data is obfuscated while it is being entered and only generic "Access-denied" messages are provided for invalid usernames and passwords.

5.3.4 FIA_X509_EXT.1: X509 Certificate Validation

The TOE validates certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- Validation of a certification path is performed 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 a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3.

- The TOE validates the extendedKeyUsage field according to the following rules[2]:
  - 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.
- The TOE only treats a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Certificate revocation checking using CRL is performed for communications with the syslog server and when configured for assets supporting TLS server certificates. See Section 8 for the assets that can be configured to use TLS. Uploading a signed CSR response also has verification of the trust chain, including revocation checking. A CRL check is done when a certificate needs to be validated and is performed for each certificate in the path presented for the aforementioned channels.

5.3.5 FIA_X509_EXT.2: X509 Certificate Authentication

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication of TLS servers in the operational environment.

Validation checking can be enabled or disabled for each asset that supports TLS server certificate validation. When validation checking is enabled and the TOE cannot establish a connection to determine the validity of a certificate, the TOE will not accept the certificate and will not establish the connection.

When the TOE connects to an asset that has the Verify SSL Certificate option enabled, the TOE compares the signing authority of the certificate presented to the certificates in the trusted certificate store.

The root CA certificates and intermediate CA certificates used for validating a presented certificate can be added to or removed from the TOE's trust store from the **`Administrative Tools | Settings | Certificates | Trusted Certificates`** tab (as described in the administrative guide *Trusted Certificates* section).

5.3.6 FIA_X509_EXT.3: X509 Certificate Requests

The TOE generates Certificate Signing Requests as specified by RFC 2986 and is able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country. The TOE validates the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.4 Security Management

The TOE provides a Web UI and REST API management interfaces that an administrator can access via a network port. Administrators can use the desktop client to access the API or the API can be accessed directly using a web browser. Local management of the TOE is possible by directly connecting the

[2] The TOE does not support the use of OCSP or certificates for trusted updates or for executable code integrity verification.

administrator's laptop to the appliance's XO port via Ethernet cable. The management interfaces are protected with HTTP over TLS (HTTPS) and are limited to the authorized administrator as defined by their assigned role. The TOE provides security management functions and defines roles that can be associated with users in order to manage the TOE locally or remotely.

5.4.1 FMT_MOF.1/Functions: Management of Security Functions Behavior

The TOE restricts the Audit Log Management function and the functions to configure syslog servers to Security Administrators.

The Appliance admin can configure the TOE to send audit records to a syslog server from the Web UI using the command: **Administrative Tools | Settings | External Integration | Syslog**.

The Appliance admin can configure audit retention period, which causes oldest logs to be deleted and/or archived. The **`Audit Log Management`** tab allows the administrator to define and schedule an audit log management task to purge audit logs from the TOE. The frequency can be configured with or without start and end times by minutes, hours, days, or months. If the audit logs are being deleted from the appliance and not backed up on an archive server, then configuration is restricted to entering the number of days in the **`Delete audit logs older than __ days`** command**.**

5.4.2 FMT_MOF.1/ManualUpdate: Management of Security Functions Behavior

The TOE restricts the ability to enable the manual updates function to Security Administrators.

5.4.3 FMT_MTD.1/CoreData: Management of TSF Data

The TOE restricts the ability to manage the TSF data to Security Administrators.

5.4.4 FMT_MTD.1/ CryptoKeys: Management of TSF Data

The TOE restricts the ability to manage the cryptographic keys to Security Administrators. The TOE provides interfaces to import, create, add, and remove server certificate; and to add and remove TLS and trusted certificates. Certificates can be managed from the UI's **Administrative Tools | Settings | Certificates** page.

5.4.5 FMT_SMF.1: Specification of Management Functions

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;
- Ability to update the TOE, and to verify the updates using a digital signature capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to start and stop services;

- Ability to configure audit behavior (i.e. configure audit retention period which causes oldest logs to be deleted);
- Ability to modify the behaviour of the transmission of audit data to an external IT entity (i.e. configure the TOE to transmit local audit information to an external syslog);
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors; and
- Ability to import X.509v3 certificates to the TOE's trust store.

Administration is possible through the Web UI and REST API and can be accessed both remotely over a network or locally through direct connection and are protected with HTTP over TLS (HTTPS). Local management of the TOE can be performed by directly connecting the administrator's laptop to the appliance's XO port via Ethernet cable however the TOE is intended to be managed remotely from a remote HTTPS/TLS client. The ability to set system time is provided by the API. Otherwise, all management functions are available from either the UI or the API.

5.4.6 FMT_SMR.2: Restrictions on Security Roles

The TOE provides the following default roles:

- Authorizer – Allows the user to grant permission to other users. Can also configure time zone.
- User (Admin) – Allow the user to create new users, unlock and reset passwords for non-administrative users
- Help Desk – Allow the user to unlock and set passwords for non-administrative users
- Appliance – Allow the user to edit and update the appliance, syslog, audit retention period and other basic appliance settings
- Operations – Allow the user to reboot and monitor the appliance
- Auditor – Allow the user read-only access
- Asset – Allow the user to add, edit, and delete partitions, assets, and accounts. Can delegate authority to manage the assets to non-admin users.
- Security Policy – Allow the user to add, edit, and delete entitlements and policies that control access to accounts and assets

A factory default pre-defined Bootstrap Administrator account is used for initial configuration. The account's default password is changed at initial start-up and the account is used to create another Authorized administrator. After this, the Bootstrap Administrator account is disabled.

A user can be assigned to multiple roles. All roles except Operations and Auditor are considered to be the Security Administrator role as defined by the PP.

5.5 Protection of the TSF

The TOE implements features designed to protect itself to ensure the reliability and integrity of its security features to include protecting sensitive data; and providing timing mechanisms to ensure that reliable time information is available (e.g., for log accountability).

The TOE includes functions to perform self-tests so that it can detect when it is failing and transition to a secure, quarantine state. It also includes a mechanism to verify TOE updates to prevent malicious or other unexpected changes in the TOE.

5.5.1 FPT_APW_EXT.1: Protection of Administrator Passwords

Administrator passwords are stored as Password-Based Key Derivation Function 2 (PBKDF2) hashes in a data store. The TOE uses a SHA-512 hash algorithm with 1000 iterations to generate a 256-bit key. There are no administrative interfaces to view the stored password data.

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

The TOE does not provide interfaces that allow pre-shared, symmetric or private keys to be read. The Data Protection API (DPAPI) encrypts the persisted key data using AES-256.

5.5.3 FPT_STM_EXT.1: Reliable Time Stamps

The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE's embedded OS manages the clock and exposes administrator clock-related functions such as set time. The clock is used for audit record time stamps, measuring session activity for termination, and for cryptographic operations based on time/date. The TOE provides reliable time stamps for its own use and allows the Security Administrator to set the time using the API (**`SystemTime`**). Additionally, the time zone can be configured using: **`Administrative Tools | Settings | Safeguard Access | Time Zone`**.

5.5.4 FPT_TST_EXT.1: TSF Self-Testing

During initial start-up (on power on), the TOE runs a suite of the following self-tests: Known Answer Tests (KAT), and Firmware Integrity tests (bootchain executable code integrity). A known-answer test involves operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test fails. The Firmware Integrity Test uses a 2048-bit RSA hardware-protected asymmetric key and SHA-256 hash (TPM) to verify integrity. If the calculated result does not equal the previously generated result, the software/firmware test will fail.

The Kernel Mode Cryptographic Primitives Library automatically performs the following self-tests upon startup:

- HMAC (SHA-1, SHA-256, and SHA-512) Known Answer Tests
- AES-128 encrypt/decrypt CBC Known Answer Tests
- AES-128 encrypt/decrypt GCM Known Answer Tests

- RSA sign/verify Known Answer Tests using RSA_SHA256_PKCS1 signature generation and verification
- ECDSA sign/verify Known Answer Tests on P256 curve
- DH secret agreement Known Answer Test with 2048-bit key
- ECDH secret agreement Known Answer Test on P256 curve
- SP 800-90A AES-256 counter mode DRBG Known Answer Tests (instantiate, generate and reseed)
- SP 800-108 KDF Known Answer Test
- SP 800-132 PBKDF Known Answer Test

The TOE performs bootchain and executable code integrity verification with digital signature using hardware-protected asymmetric key and hash (TPM). The TOE includes a Microsoft Windows 10 IoT Enterprise OS BitLocker with TPM that performs full disk encryption. The TOE uses Microsoft's Secure Boot function included with the OS to check the boot loader before launching it and ensure it is signed by Microsoft. If the signature is validated, Bitlocker in conjunction with the TPM, validates the integrity of the boot and system files before decrypting them; an unsuccessful validation will prohibit access to a protected system. The bootchain tests (Firmware Integrity tests) are performed using the TPM platform configuration registers (PCRs).

Integrity measurements are performed on TPM's Platform Configuration Registers (PCR) using SRTM (Static Root of Trust for Measurements) at system boot. The measurements are performed as follows. The first thing executed at boot is called the Core Root of Trust for Measurements (CRTM) aka the BIOS boot block which measures the BIOS and send the value (hash) to the TPM PCR 0 location before executing it. Then the BIOS measures the next item in the boot chain and again, will store the value in a PCR of the TPM. This process is executed for each of the following TPM PCRs:

- PCR 0 : Core Root of Trust Measurement (CRTM), BIOS and Platform Extensions
- PCR 2 : Option ROM Code
- PCR 4 : Master Boot Record (MBR) Code
- PCR 8 : NTFS Boot Sector
- PCR 9 : NTFS Boot Block
- PCR 10 : Boot Manager
- PCR 11 : BitLocker Access Control

By measuring the state of each of the components in the boot sequence, boot and system file integrity can be assured. If validation is successful, then the protected files are decrypted. If the validation is unsuccessful then the files are not decrypted and access is prohibited. The SHA-256 hash is used to validate each part of the bootchain.

If any of the self-tests fail, the TOE will enter quarantine mode (i.e., no longer in the evaluated configuration). The TOE enters an error state and outputs an error indicator. The TOE will not boot and does not perform any cryptographic operations while in the error state. All data output from the TOE is inhibited when an error state exists. If this occurs, the appliance should be re-booted.

The self-tests are sufficient to demonstrate that the TSF is operating correctly since they encompass the cryptographic functionality and the integrity of the entire TOE firmware executable code. In addition, since the disk is encrypted it cannot be tampered with while unpowered.

5.5.5 FPT_TUD_EXT.1: Trusted Update

The TOE provides Security Administrators with a manual trusted update mechanism used to initiate updates to TOE firmware. The firmware image signed by One Identity using RSA 4096 is downloaded from the One Identity support website by an administrator. The TOE performs the signature check when the update is first uploaded to the TOE and will be rejected if the check fails. The TOE will only install an update that has passed the signature validation.

Administrators can query the currently executing TOE version via the API from the **`Appliance Information`** tab or from the **`Settings: Appliance`** page on the web UI.

5.6 TOE Access

The TOE displays a Security Administrator-specified advisory notice and consent-warning message prior to establishing an administrative user session. The TOE terminates local and remote administrator interactive sessions after a Security Administrator-specified time period of inactivity. The TOE allows administrator-initiated termination of the administrator's own interactive session.

5.6.1 FTA_SSL_EXT.1: TSF-Initiated Session Locking

The TOE terminates local interactive sessions after a Security Administrator-specified time period of inactivity between 5 minutes and 2880 minutes. The default is 15 minutes. This function applies to the web UI. The REST API is not an interactive session and therefore does not maintain the concept of idleness. When REST API access is granted, the authorization is active for a configurable period of time between 10 minutes and 20 days.

5.6.2 FTA_SSL.3: TSF-Initiated Termination

The TOE terminates remote interactive sessions after a Security Administrator-specified time period of inactivity between 5 minutes and 2880 minutes. The default is 15 minutes. This function applies to the web UI. The Desktop Client API is not subject to the termination feature since it is a REST API requiring the token to be included with each command and does not constitute an interactive session as defined in the PP.

5.6.3 FTA_SSL.4: User-Initiated Termination

The TOE allows administrator-initiated termination of the administrator's own interactive session with the web UI logout option. This function applies only to web UI since the API is not considered an interactive session as defined in the PP.

5.6.4 FTA_TAB.1: Default TOE Access Banners

Before establishing an administrative user session, the TOE display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE for each of the Web UI and REST API (HTTPS) management interfaces.

5.7 Trusted Path/Channels

The TOE provides trusted paths and channels for remote administrators and trusted IT entities. The TOE can be configured to send audit records to external syslog server(s) using TLS in real-time.

5.7.1 FTP_ITC.1: Inter-TSF Trusted Channel

The TOE supports the use of trusted communication channels between itself and authorized IT entities for assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

The TOE can be configured to send audit records to external syslog server(s) using TLS in real-time. The TOE permits the TSF to initiate communication for remote audit storage and password management using the following:

- Syslog server using TLS, and
- Password management (connections with assets) using TLS, or HTTPS (see Section 8).

The TOE communicates with its authorized trusted entities over TLS or HTTPS. All communications are sent over the trusted channel and are protected by the security protocols. The underlying cryptographic algorithms and implementation are CAVP-validated and are part of the TOE.

5.7.2 FTP_TRP.1: Trusted Path

The TOE provides HTTPS (TLSv1.2) to support secure remote administration when administrators access the web UI and REST APIs remotely. Administrators can initiate a remote session that is secured (from disclosure and modification) using CAVP-validated cryptographic operations. All remote security management functions require the use of a secure channel.

6 Protection Profile Claims

The ST conforms to the collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020, [CPP_ND_V2.2E].

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [CPP_ND_V2.2E] has been included by reference into this ST.

As explained in Section 3, Security Objectives, the Security Objectives of the [CPP_ND_V2.2E] have been included by reference into this ST.

All Security Functional Requirements (SFRs) in this ST have been reproduced from the [CPP_ND_V2.2E] and operations completed as appropriate.

7 Rationale

This security target includes by reference the [CPP_ND_V2.2E] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [CPP_ND_V2.2E] assumptions. [CPP_ND_V2.2E] security functional requirements have been reproduced with the Protection Profile operations completed. Operations on the security requirements follow [CPP_ND_V2.2E] application notes and assurance activities. The security target did not add or remove any security requirements. Consequently, [CPP_ND_V2.2E] rationale applies and is complete.

8 Supported Assets

This section identifies the TOE supported assets and the trusted communication methods supported for each: TLS or HTTPS (green highlighting). Some asset types do not use encrypted channels; these have been omitted as they are outside the scope of the TOE. Please see One Identity Safeguard for Privileged Passwords 6.7 Administrator Guide, Table 8 for a full list of supported assets to include those that are outside of the evaluation.

Asset Type	TLS	HTTPS
ACF2 - Mainframe r14 zSeries	✓	X
ACF2 - Mainframe r15 zSeries	✓	X
ACF2 - Mainframe LDAP r15 zSeries	✓	X
ACF2 - Mainframe LDAP r14 zSeries	✓	X
Amazon Web Services 1	X	✓
ESXi 5.5	X	✓
ESXi 6.0	X	✓
ESXi 6.5	X	✓
ESXi 6.7	X	✓
Facebook (Deprecated)	X	✓
IBM i 7.1 PPC	✓	X
IBM i 7.2 PPC	✓	X
IBM i 7.3 PPC	✓	X
MongoDB 3.4	✓	X
MongoDB 3.6	✓	X

Asset Type	TLS	HTTPS
MongoDB 4.0	✓	X
MySQL 5.6	✓	X
MySQL 5.7	✓	X
OpenLDAP 2.4	✓	X
Oracle 11g Release 2	✓	X
Oracle 12c Release 1	✓	X
PostgreSQL 9.6 Other	✓	X
PostgreSQL 10.2 Other	✓	X
PostgreSQL 10.3 Other	✓	X
PostgreSQL 10.4 Other	✓	X
PostgreSQL 10.5 Other	✓	X
RACF - Mainframe z/OS V2.1 Security Server zSeries	✓	X
RACF - Mainframe z/OS V2.2 Security Server zSeries	✓	X
RACF - RACF - Mainframe LDAP z/OS V2.1 Security Server zSeries	✓	X
RACF - RACF - Mainframe LDAP z/OS V2.2 Security Server zSeries	✓	X
SAP HANA 2.0 Other	✓	X
SonicWALL SMA or CMS 11.3.0	X	✓
SQL Server 2012	✓	X
SQL Server 2014	✓	X
SQL Server 2016	✓	X
Sybase (Adaptive Server Enterprise) 15.7	✓	X
Sybase (Adaptive Server Enterprise) 16	✓	X
Top Secret - Mainframe r14 zSeries	✓	X

Asset Type	TLS	HTTPS
Top Secret - Mainframe r15 zSeries	✓	X
Top Secret - Mainframe LDAP r14 zSeries	✓	X
Top Secret - Mainframe LDAP r15 zSeries	✓	X
Twitter (Deprecated)	X	✓